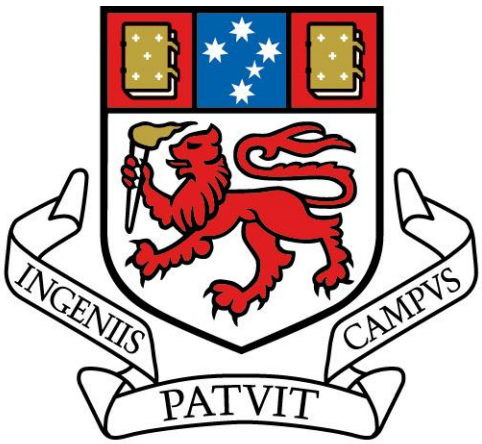


New Evolutionary Algorithms for Mining Interesting Association Rules

Mir Md. Jahangir Kabir

M.Sc. in Information Technology, University of Stuttgart, Germany

B.Sc. in Computer Science & Engineering, Rajshahi University of Engineering and Technology, Bangladesh

Submitted in fulfilment of the requirements for the degree of Doctorate of Philosophy at the School of Engineering and ICT, University of Tasmania (May, 2016)

Declaration

I, Mir Md. Jahangir Kabir, do hereby declare that this thesis contains no material that has been accepted for the award of any other degree or diploma in any tertiary institution, except by way of background information and duly acknowledged in the thesis. To the best of my knowledge and belief it contains no material previously published by another person, except where due reference is made in the text of the thesis, nor does the thesis contain any material that infringes copyright.

Signed: .....

Date: 24th May, 2016

Authority of Access

This thesis may be made available for loan and limited copying in accordance with the *Copyright Act 1968*.

Signed: .....

Date: 29th May, 2016

Statement of Co-authorship

The following people and institutions contributed to the publication of the work undertaken as part of this thesis:

Mir Md Jahangir Kabir (Candidate)
Shuxiang Xu (co-author)
Byeong Ho Kang (co-author)
Zongyuan Zhao (co-author)

All of these authors are within School of Engineering and ICT, University of Tasmania

Kabir, MMJ and Xu, S and Kang, BH and Zhao, Z, "A New Multiple Seeds Based Genetic Algorithm for Discovering a Set of Interesting Boolean Association Rules", (Submitted to a Journal, 2016).

Mr. Mir Md. Jahangir Kabir (70%) is the primary author. He conducted the research and prepared the material for publication. Dr. Shuxiang Xu (15%), Dr. Byeong Ho Kang (10%) and Mr. Zongyuan Zhao (5%) of the School of Computing and Information Systems, University of Tasmania, all of them provided general guidance and editing as supervisors and colleague.

Kabir, MMJ and Xu, S and Kang, BH and Zhao, Z, "Multiple Seeds Based Evolutionary Algorithm for Mining Boolean Association Rules", *5th PAKDD Workshop on Biologically Inspired Data Mining Techniques*, 19-22 April, 2016, Auckland, New Zealand (In press).

Mr. Mir Md. Jahangir Kabir (70%) is the primary author. He conducted the research and prepared the material for publication. Dr. Shuxiang Xu (15%), Dr. Byeong Ho Kang (10%) and Mr. Zongyuan Zhao (5%) of the School of Computing and Information Systems, University of Tasmania, all of them provided general guidance and editing as supervisors and colleague.

Kabir, MMJ and Xu, S and Kang, BH and Zhao, Z, "Discovery of interesting association rules using genetic algorithm with adaptive mutation", *Neural Information Processing 22nd International Conference, ICONIP 2015, Proceedings, Part II*, 09-12 November, Istanbul, Turkey, pp. 96-105. ISBN 978-3-319-26534-6 (2015).

Mr. Mir Md. Jahangir Kabir (70%) is the primary author. He conducted the research and prepared the material for publication. Dr. Shuxiang Xu (15%), Dr. Byeong Ho Kang (10%) and Mr. Zongyuan Zhao (5%) of the School of Computing and Information Systems, University of Tasmania, all of them provided general guidance and editing as supervisors and colleague.

Kabir, MMJ and Xu, S and Kang, BH and Zhao, Z, "A new evolutionary algorithm for extracting a reduced set of interesting association rules", *Neural Information Processing 22nd International Conference, ICONIP 2015, Proceedings, Part II*, 09-12 November, Istanbul, Turkey, pp. 133-142. ISBN 978-3-319-26534-6 (2015).

Mr. Mir Md. Jahangir Kabir (70%) is the primary author. He conducted the research and prepared the material for publication. Dr. Shuxiang Xu (15%), Dr. Byeong Ho Kang (10%) and Mr. Zongyuan Zhao (5%) of the School of Computing and Information Systems, University of Tasmania, all of them provided general guidance and editing as supervisors and colleague.

Kabir, MMJ and Xu, S and Kang, BH and Zhao, Z, "Comparative analysis of genetic based approach and apriori algorithm for mining maximal frequent item sets", *Proceedings of the 2015 IEEE Congress on Evolutionary Computation*, 25-28 May 2015, Sendai, Japan, pp. 39-45. ISBN 978-1-4799-7492-4 (2015).

Mr. Mir Md. Jahangir Kabir (70%) is the primary author. He conducted the research and prepared the material for publication. Dr. Shuxiang Xu (15%), Dr. Byeong Ho Kang (10%) and Mr. Zongyuan Zhao (5%) of the School of Computing and Information Systems, University of Tasmania, all of them provided general guidance and editing as supervisors and colleague.

Kabir, MMJ and Xu, S and Kang, BH and Zhao, Z, "GeneticMax: An Efficient Approach to Mining Maximal Frequent Itemsets Based on Genetic Algorithms", *IT in Industry*, **3** (3) pp. 64-73. ISSN 2204-0595 (2015).

Mr. Mir Md. Jahangir Kabir (70%) is the primary author. He conducted the research and prepared the material for publication. Dr. Shuxiang Xu (15%), Dr. Byeong Ho Kang (10%) and Mr. Zongyuan Zhao (5%) of the School of Computing and Information Systems, University of

Tasmania, all of them provided general guidance and editing as supervisors and colleague.

Kabir, MMJ and Xu, S and Kang, BH and Zhao, Z, "A novel approach to mining maximal frequent itemsets based on genetic algorithm", *Proceedings of the 9th International Conference on Information Technology and Applications*, 1-4 July 2014, Sydney, Australia, pp. 1-6. ISBN 978-0-9803267-6-5 (2014).

Mr. Mir Md. Jahangir Kabir (70%) is the primary author. He conducted the research and prepared the material for publication. Dr. Shuxiang Xu (15%), Dr. Byeong Ho Kang (10%) and Mr. Zongyuan Zhao (5%) of the School of Computing and Information Systems, University of Tasmania, all of them provided general guidance and editing as supervisors and colleague.

Kabir, MMJ and Xu, S and Kang, BH and Zhao, Z, "Association rule mining for both frequent and infrequent items using particle swarm optimization algorithm", *International Journal on Computer Science and Engineering*, 6 (7) pp. 221-231. ISSN 0975-3397 (2014).

Mr. Mir Md. Jahangir Kabir (70%) is the primary author. He conducted the research and prepared the material for publication. Dr. Shuxiang Xu (15%), Dr. Byeong Ho Kang (10%) and Mr. Zongyuan Zhao (5%) of the School of Computing and Information Systems, University of Tasmania, all of them provided general guidance and editing as supervisors and colleague.

Kabir, MMJ and Xu, S and Kang, BH and Zhao, Z, "A hybrid GeneticMax algorithm for improving the traditional genetic based approach for mining maximal frequent item sets", *International Journal of Computer Science and Network Security*, **14** (10) pp. 27-35. ISSN 1738-7906 (2014).

Mr. Mir Md. Jahangir Kabir (70%) is the primary author. He conducted the research and prepared the material for publication. Dr. Shuxiang Xu (15%), Dr. Byeong Ho Kang (10%) and Mr. Zongyuan Zhao (5%) of the School of Computing and Information Systems, University of Tasmania, all of them provided general guidance and editing as supervisors and colleague.

We the undersigned agree with the above stated "proportion of work undertaken" for each of the above published (or submitted) peer-reviewed manuscripts contributing to this thesis:

Signed: _____

Dr Shuxiang Xu
Supervisor
School of Engineering & ICT
University of Tasmania
Date: 13/04/2016

Prof Andrew Chan
Head of School
School of Computing & ICT
University of Tasmania
Date: 20/4/2016

Abstract

This PHD thesis deals with the evolutionary algorithms for mining frequent patterns and discovering useful and interesting Boolean association rules from large data sets. Initially, the classical algorithms for mining frequent patterns and single and multi- objective evolutionary algorithms for discovering association rules using different measures are studied. Secondly, the problem of extracting frequent patterns using classical algorithms and obtaining a set of high quality association rules relying on the evolutionary algorithms are addressed.

The objectives of this thesis are as follows:

1. Designing evolutionary algorithms for extracting frequent patterns from large data sets.
2. Designing multi-objective evolutionary algorithms for discovering a reduced set of high quality Boolean association rules from categorical data sets.
3. Improving the single seed based genetic algorithm by designing a multiple seeds based genetic algorithm for mining Boolean association rules.

To accomplish these objectives, this research evolved different evolutionary algorithms for mining frequent patterns efficiently, and obtaining high quality Boolean association rules (BARs).

Firstly, the method named GeneticMax, a new approach based on a genetic algorithm, is used to mine maximal frequent item sets by accessing a large data set for fewer number of nodes. This method is improved by another approach named Hybrid GeneticMax. This new model which outperforms the GeneticMax algorithm if there are a reasonable amount of infrequent items in 1- item sets. This proposal shows the power of using an evolutionary algorithm along with a local search mechanism for generating maximal frequent item sets from a lexicographic tree. On the other hand, this research proposed particle swarm optimization (PSO) based approach, a new heuristic algorithm for mining association rules for both frequent and infrequent items. This approach can mine rules for more than three items.

Secondly, a new multi-objective evolutionary model named Association Rules Mining with Genetic Algorithm Using an Adaptive Mutation Method (ARMGAAM), which is

very useful for mining reduced sets of Boolean association rules from categorical data sets. Another method named Mining Boolean Association Rules with Evolutionary Algorithm (MBAREA), a new evolutionary model which extends the existing Association Rule Mining with Genetic Algorithm (ARMGA) and Multi-objective Association Rule Mining with Genetic Algorithm (ARMMGA), maximizes two objectives; performance and interestingness. The former method uses a re-initialization technique along with an adaptive mutation method whereas the latter uses a class based mutation method along with a best population technique. Both methods discover a reduced set of BARs from different data sets with a good trade-off among the number of generated rules and different measures.

Finally, MSGA, a new genetic algorithm based on multiple seeds for producing an effective initial population, has a higher search efficiency along with good convergence speed, prevents the limitation of selecting an effective single seed for generating an initial population for mining BARs. Of particular note, the selection of above mentioned evolutionary algorithms depends on the specific needs of users.

Acknowledgements

I would like to express my sincere gratitude to those who gave me the continuous support and cooperation during my PhD study.

First and foremost, all praise goes to the almighty Allah who gave me the patience for completing this PhD thesis.

I would like to express my deepest and sincere gratitude to my thesis supervisors, Dr. Shuxiang Xu and Professor Byeong Ho Kang. Dr. Shuxiang had spent dedicated time for providing constructive and challenging feedback to improve my research work. I am also grateful to Professor Byeong for his valuable suggestions and support. His advice helped me to improve my research skills. It was not possible to complete this thesis without their help, motivation, support, encouragement, and guidance.

I would like to convey my heartfelt gratitude to the graduate research coordinator, Dr. Leonie Ellis for her valuable advice and supports through constructive review of this research work.

I am grateful for the International Postgraduate Research Scholarship (IPRS), Tasmania Graduate Research Scholarship (TGRS), and School of Engineering and ICT and Faculty of Science, Engineering and Technology at the University of Tasmania for their financial support for the past three years.

I wish to thank Dr. Mark Brown and Amanda Lunt for proof reading of every chapters of my thesis. Thank you to my office mate Zongyuan Zhao and Dr. Simon for their help and delicious food.

A special gratitude and love goes to my parents for their inspiration, guidance, unconditional love, and support which helped me to complete the thesis successfully.

Last but not least, I want to express my deepest love and thanks to my wife and son for their unconditional love, encouragement and support.

Table of Contents

List of Figures

List of Tables

Chapter 1 - Introduction

1.1 Introduction

The brief overview of this research is introduced by this chapter. It explains the problem statements of data mining techniques and the motivation in section 1.2 and 1.3, respectively. The major contributions of this thesis are discussed through section 1.4. The synopsis of the thesis chapters is presented as an outline in section 1.5. Finally, the data sets are discussed in section 1.6.

1.2 Problem Statement

Data mining is one of the fundamental research areas in artificial intelligence. Association rule mining plays a vital role in advancing the research, applications and development of data mining techniques. The generation of data in different research areas introduces a new set of opportunities and challenges in the way of searching and retrieving information. Data mining techniques need to handle a large volume of data for analysis. Thus, in the last few years a large number of research papers are dedicated to data mining research areas. Data mining techniques, or knowledge discovery in databases (KDD) define the extraction of novel, valid, interesting, useful, and understandable knowledge or patterns from data (Fayyad et al. 1996). Knowledge can be learned from the past experiences of users or it can be obtained from stored data. For example, a doctor listens to the symptoms from a patient, diagnoses the disease and prescribes an appropriate treatment based on his knowledge of medical science. A very common real life application is shopping market basket analysis, in which retailers seek to understand the purchase behaviour of customers.

The data analysis attempts to find interesting hidden relationships among products purchased by customers through association rule mining/frequent pattern mining tasks. For example, by using a frequent pattern mining task, a shop manager may discover that butter, milk and bread are frequently purchased together by customers. In another example, one association rule may indicate that when a customer buys coffee he would also buy milk. Such information can then be used for purposes of cross-selling and upselling, in addition to influencing sales promotions, store design, and discount plans. Similarly, a video shop manager can use frequent pattern mining/ association rule mining to recommend related videos or games when a customer has hired or bought a specific video or game, so that the customer would come back. Web administrators can use frequent patterns or association rules to understand particular collections of web pages

which are viewed together by a group of web users. These sorts of interesting relationships (e.g., correlation, association) among data items help managers make relevant policies within their industries. For this reason discovering knowledge from various datasets is used to solve different complex problems of real world applications. Different kinds of knowledge are generated by data mining approaches such as association rules, cluster, and classification rules and so on.

Association rule mining tasks include two steps: first, mining frequent item sets from a large database, and second, generating association rules or correlation relationship among a large set of data items. Nowadays, huge amounts of data are collected and stored by industries, who are interested in mining frequent item sets from large databases. The discovery of association rules among large amount of business transactions helps industries make business decisions (Russell & Norvig 2008, pp. 16-27; Wang et al. 2003). The problem is formulated as follows: a set of items and a large collection of transactions have been given, each transaction is a subset of these items, find all frequent item sets. The number of frequent item sets is defined by a user specified percentage value of the database.

Let $D = \{t_1, t_2, t_3, \ldots\ldots, t_k, \ldots t_n\}$ be the database or data set, where $t_1, t_2, \ldots\ldots, t_n$ are the n number of transactions in the database. Each transaction t_i is a set of items $I = \{i_1, i_2, \ldots\ldots, i_k, \ldots i_n\}$, where i_1 is item number 1, i_2 is item number 2 and so on. Transaction $t_i$is represented as a binary vector. If $t_i[k] = 1$ then it means that t_i bought the item i_k , otherwise $t_i[k] = 0$. Let X be a set of few items in I i.e. $X \subseteq I$ .The set $t_i(X) \subseteq I$ is true for all items in itemset X for transaction t_i. The support value of an item is how many times the item appears in the transaction database as a subset (Yan et al. 2009; Jesus et al. 2011). The support value of an item set is denoted by $\sigma(X) = |\{t_1(X) + t_2(X) + t_3(X) \ldots\ldots\ldots t_{n-1}(X) + t_n(X)\}|/|D|$. Here $t_n(X)$ gives the binary value. If the examined itemset X appears as a subset in a transaction t_i, then $t_i(X)$ = 1, otherwise $t_i(X)$= 0. An item set with 1 item is called a 1-item set, an item set with k-items is called a k-item set. An item set is called frequent if its support value is more than or equal to a user defined threshold value, which is denoted by min_supp (minimum support) i.e. $\sigma(X) \geq$ min_supp. Frequent item sets are denoted by FI. If an item set X is frequent and no superset of X is frequent then X is a maximal frequent item set and the set

of all maximal frequent item sets are denoted by MFI (Agrawal & Srikant 1994; Agrawal et al. 1993; Agarwal et al. 2001).

The problem of mining association rules has been considered by many researchers and a large number of algorithms have been developed for extracting association rules from different type of databases (Agrawal et al. 1993; Aggarwal & Yu 1998; Agrawal & Srikant 1994; Silverstein et al. 1998; Agrawal & Shafer 1996; Wu et al. 2004). An association rule, $A \rightarrow B$, is an implication between two item sets A and B. Most of the existing association rule mining algorithms are based on a support-confidence framework. This framework consists of two sub processes and most of the existing association rule mining algorithms follow these factors to measure the interestingness of a rule. These factors are as follows (Agrawal & Srikant 1994; Jesus et al. 2011; Hipp et al. 2000; Yan et al. 2009):

1) finding all frequent item sets from a large database which satisfy a user defined support value and
2) generating rules from those frequent item sets which satisfy a user defined confidence value.

For example, clock and battery are the products in a shopping centre. A rule, clock→battery, with its support value being 0.1 and confidence 0.50, means that in total there are 10% of transactions containing both clock and battery whereas 50% of transactions containing clock also contain battery.

Although this from the following major problems (Yan et al. 2009; Yan et al. 2005; Zhou & Yau 2007):

1) Users need to specify an appropriate threshold value although they have no knowledge regarding the database.
2) Satisfying a minimum support value reveals an exponential search space of 2^n, where n is the number of item sets. Finally, it may generate a huge number of unnecessary rules from frequent item sets.

1.3 Motivation

As discussed in section 1.2, the traditional association rule mining task considers two steps. In first step, it finds frequent patterns from a large database based on a user de-

fined support value and in second step it generates rules from frequent patterns which satisfies a user defined confidence value. That is, a rule is valid if it satisfies user defined minimum support and confidence values. Therefore, users need to specify these two threshold values for their mining job although they have no knowledge about the database.

Lots of research papers have been published for measuring the interestingness of a rule. To measure the interestingness of a rule, $A \rightarrow B$, researchers (Piatetsky-Shapiro 1991) used the rule interest, RI = P(A,B) - P(A)P(B) as a constraint. For finding correlated association patterns, the chi-square test is used by the researchers (Silverstein et al. 1998). Based on a probability ratio, another interesting model is proposed by the researchers (Wu et al. 2002). Wu et al. developed their methods for efficient searching of positive and negative association rules by using a constraint function based on minimum support, confidence and the Piatetsky- Shapiro based rule interest (Wu et al. 2004). These algorithms raise the following major challenges (Yan et al. 2009; Yan et al. 2005; Zhou & Yau 2007):

3) Users need to specify an appropriate threshold value for mining rules although they have no information regarding the database.
4) Association rule mining is an NP-Hard problem because searching all frequent item sets satisfying a minimum support value reveal an exponential search space of 2^n, where n is the number of item sets. Finally, it may generate a huge number of unnecessary rules from frequent item sets, resulting in weak mining performance.

To avoid these problems researchers used genetic algorithm based approaches because a genetic algorithm is an efficient tool for a global search, especially when the search space is too large to use deterministic search methods (Mukhopadhyay et al. 2014). In evolutionary algorithm based approaches, users do not need to specify the threshold value explicitly because it imitates the natural evolution process along with genetic operators such as selection, crossover and mutation. A large number of research papers are dedicated to mine Boolean/ categorical (Yan et al. 2005; Yan et al. 2009; Wakabi-Waiswa & Baryamureeba 2008; Shenoy et al. 2003; Shenoy et al. 2005; Qodmanan et al. 2011), quantitative/numerical (Salleb-aouissi et al. 2013; Martin et al. 2014; Alatas & Akin 2008a) and fuzzy association rules (Kaya & Alhajj 2006; Kaya & Alhajj 2005;

Hong et al. 2008) using genetic algorithm based approaches (Jesus et al. 2011). In this research, Boolean association rules (BARs) are considered.

Wakabi-Waiswa and Baryamureeba (Wakabi-Waiswa & Baryamureeba 2008) proposed a Pareto based multi-objective evolutionary algorithm to mine interesting association rules instead of generating unknown numbers of unnecessary rules which is done by traditional mining algorithms. They used J-measure and perplexity along with other metrics such as comprehensibility, interestingness and predictive accuracy which are used to improve the interestingness of association rules. To keep the interesting rules which are generated at some point in intermediate population generation, in this approach researchers used an external population which indicates an extra overhead for this method. Another genetic algorithm based model for measuring the interestingness of rules is given by ARMGA (Yan et al. 2005) and EARMGA (Yan et al. 2009). Instead of using a support-confidence framework, these algorithms used conditional probability as a fitness function which incorporates Pitatesky- Shapiro's rule interest method. Experimental results show that, a large number of high quality rules as well as unnecessary rules are generated by classical and evolutionary algorithm based approaches due to considering weak fitness function (Martin et al. 2014; Kabir et al. 2015a). Because of the use of simple genetic operators like mutation, these approaches missed some high quality rules which are generated in intermediate generation of a population.

Most of the association rule mining algorithms use a single seed for generating an initial population. Researchers show that initial populations have significant effects of producing good solutions over several generations (Maaranen et al. 2007). Single seed based evolutionary algorithms suffer from the following major challenges:

1) Different seed chromosomes generate different initial populations. Because of this reason different seed chromosomes yield different results.
2) It is a hard process to define a good seed for a specific application.
3) Defining seed is not an automatic process rather it is manual since the maximum range of a seed chromosome varies from data set to data set. For example, data sets A and B contain 100 and 50 items respectively. So, the range of a gene of a chromosome for a data set A, should be in between 1 to 100. On the other hand, for a data set B it should be 1 to 50.

To avoid these problems, the following techniques are incorporated by the proposed approaches:

1) Developing mutation operators along with best population and re-initialization techniques to avoid the generation of iterative rules and keep the high quality rules which are generated in the intermediate generation of a population.
2) Strengthen the fitness function by using different measures such as minimum interest, lift, conditional probability and so on.
3) Subdivide the whole solution space into *m*-domain to get a seed from each domain. Finally, these seeds will be used to generate an effective initial population.

The research questions of this thesis are:

1) What is required for designing new evolutionary algorithms for mining maximal frequent item sets efficiently?
2) Which mechanisms are used for designing new multi-objective evolutionary algorithms for discovering a reduced set of high quality Boolean association rules?
3) What are the techniques by which an effective initial population is generated for further evolution based on multiple seeds?

The specific objectives of this research are as follows:

1) Designing evolutionary algorithms for extracting frequent patterns from large data sets.
2) Designing multi-objective evolutionary algorithms for discovering a reduced set of high quality Boolean association rules from categorical data sets.
3) Improving the single seed based genetic algorithm by designing a multiple seeds based genetic algorithm for mining Boolean association rules.

1.4 Major Contributions

There are lots of demanding issues in association rule mining research. Those are classified as mining high dimensional datasets, mining interesting association rules, designing methods for scalability of large datasets, analyses of DNA sequence and so on. Evolutionary algorithms based approaches play an important role in association rule mining tasks. The following research contributions are made:

1) This thesis proposes a new approach based on a genetic algorithm to generate maximal frequent item sets (MFIs) from large datasets. This new algorithm, GeneticMax, is heuristic which mimics natural selection approaches for finding MFIs in an efficient way. This algorithm uses a lexicographic tree and avoids level by level searching which reduces the time required to mine the MFIs in a linear way. The significant contribution of this research is that it generates frequent item sets by the approach based on a genetic algorithm is scale independent to the size of the datasets. The search strategy of this new approach includes bitmap representation of the nodes in a lexicographic tree and identifying frequent item sets (FIs) from superset-subset relationships of nodes. The proposed algorithm shows how evolutionary method can be used on real datasets to find all the MFIs in an efficient way. The performance of a newly developed approach is compared against that of a famous approach named Apriori. For the experimentation of both methods, the same platform and hardware configuration are used.

   Part of this contribution has been published in:

   Kabir, MMJ and Xu, S and Kang, BH and Zhao, Z, "A novel approach to mining maximal frequent itemsets based on genetic algorithm", *Proceedings of the 9th International Conference on Information Technology and Applications*, 1-4 July 2014, Sydney, Australia, pp. 1-6. ISBN 978-0-9803267-6-5 (2014).

   Kabir, MMJ and Xu, S and Kang, BH and Zhao, Z, "GeneticMax: An Efficient Approach to Mining Maximal Frequent Itemsets Based on Genetic Algorithms", *IT in Industry*, **3** (3) pp. 64-73. ISSN 2204-0595 (2015).

   Kabir, MMJ and Xu, S and Kang, BH and Zhao, Z, "Comparative analysis of genetic based approach and apriori algorithm for mining maximal frequent item sets", *Proceedings of the 2015 IEEE Congress on Evolutionary Computation*, 25-28 May 2015, Sendai, Japan, pp. 39-45. ISBN 978-1-4799-7492-4 (2015).

2) This thesis introduces a new algorithm named, Hybrid GeneticMax, which uses a local search along with a genetic algorithm to mine maximal frequent item sets from large data sets. The aim of this new approach is converging to a solution as fast as possible, especially if 1-item sets contain a reasonable amount of infre-

quent items and the solution resides in the deep level of the lexicographic tree instead of near the root. Thorough experiments are conducted for evaluating the performance of a newly developed method with the existing one, GeneticMax, using the same platform and hardware. In addition, a new particle swarm optimization (PSO) based approach is developed for discovering the relationship among frequent items along with infrequent ones.

Part of this contribution has been published in:

Kabir, MMJ and Xu, S and Kang, BH and Zhao, Z, "A hybrid GeneticMax algorithm for improving the traditional genetic based approach for mining maximal frequent item sets", *International Journal of Computer Science and Network Security*, **14** (10) pp. 27-35. ISSN 1738-7906 (2014).

Kabir, MMJ and Xu, S and Kang, BH and Zhao, Z, "Association rule mining for both frequent and infrequent items using particle swarm optimization algorithm", *International Journal on Computer Science and Engineering*, 6 (7) pp. 221-231. ISSN 0975-3397 (2014).

3) Designing two new multi-objective evolutionary models, named Association Rules Mining with Genetic Algorithm Using an Adaptive Mutation Method (ARMGAAM) and Mining Boolean Association Rules with Evolutionary Algorithm (MBAREA), using different measures for mining a reduced set of Boolean association rules from different categorical data sets. The former method uses a re-initialization technique along with an adaptive mutation method whereas the latter uses a class based mutation method along with a best population technique. For measuring the performance of the proposed methods, the obtained results of the proposed methods are compared with existing multi-objective evolutionary algorithms and classical methods. The same platform and hardware are utilised for the experimentation.

   Part of this contribution has been published in:

   Kabir, MMJ and Xu, S and Kang, BH and Zhao, Z, "Discovery of interesting association rules using genetic algorithm with adaptive mutation", *Neural Information Processing 22nd International Conference, ICONIP 2015, Proceedings, Part II*, 09-12 November, Istanbul, Turkey, pp. 96-105. ISBN 978-3-319-26534-6 (2015).

Kabir, MMJ and Xu, S and Kang, BH and Zhao, Z, "A new evolutionary algorithm for extracting a reduced set of interesting association rules", *Neural Information Processing 22nd International Conference, ICONIP 2015, Proceedings, Part II*, 09-12 November, Istanbul, Turkey, pp. 133-142. ISBN 978-3-319-26534-6 (2015).

4) Current research shows that initial populations have significant effects of producing good solutions over several generations (Maaranen et al. 2007). Most of the association rule mining algorithms which are based on GA, use a single seed chromosome for generating an initial set of solutions. In this thesis, a new model is developed which generates multiple seeds from multiple domains of a solution space and an initial population is generated based on those seeds. The comparative analysis of this newly developed method with different single seed based algorithms with respect to different mutation and crossover operators demonstrate the efficiency of the proposed approach. For the experiment, the same hardware and platform are used for fair comparison.
   Part of this contribution has been published in:
   Kabir, MMJ and Xu, S and Kang, BH and Zhao, Z, "Multiple Seeds Based Evolutionary Algorithm for Mining Boolean Association Rules", *5th PAKDD Workshop on Biologically Inspired Data Mining Techniques*, 19-22 April, 2016, Auckland, New Zealand (In press).

   Kabir, MMJ and Xu, S and Kang, BH and Zhao, Z, "A New Multiple Seeds Based Genetic Algorithm for Discovering a Set of Interesting Boolean Association Rules", (Submitted to a Journal, 2016).

1.5 Structure of This Thesis

This section briefly introduces the thesis in the following way:

Chapter two summarizes the essential background information, including basic concepts of data mining techniques and genetic algorithms. This chapter provides a review on the literature that supports the motivation and forms the background of the thesis. Along with the intersection of data mining techniques and genetic algorithm domains, different areas within those fields, such as mining frequent patterns, extracting association rules, multi-objective evolutionary algorithms are discussed. This chapter concen-

trates on theoretical and technical background of the fields mentioned above and describes the underlying concepts with examples, when necessary.

Chapter three briefly describes the main features of the proposed methods which are used to solve the research problems. At first, the requirements for developing the frequent pattern and association rule mining tasks are described. These sections are followed by introducing two new approaches for mining frequent patterns, named GeneticMax and Hybrid GeneticMax. This chapter also discusses the method based on the particle swarm optimization technique for mining association rules. For discovering Boolean association rules from categorical data sets, new multi-objective evolutionary approaches such as adaptive and class based mutation methods are proposed. In order to address the major challenges and issues raised by a single seed based genetic algorithm, the novel features of Multiple Seeds Based Genetic Algorithm are explained. Finally, the architecture of multiple seeds based genetic algorithm is presented.

Chapter four explains the proposed methods and the implementation of the proposed approaches. The underlying concepts and structure of each approach are described. In addition, each algorithm is described by the pseudo code. Initially, the problem of mining maximal frequent item sets is addressed and the pseudo code of the GeneticMax algorithm is explained. The basic notions and the structure of the Hybrid GeneticMax algorithm are described. The underlying concept and the framework of the PSO based method for mining association rules for both frequent and infrequent items are discussed. The basic concepts, objectives and the flowchart of the proposed algorithms for mining Boolean association rules, named ARMGAAM and MBAREA, are explained, respectively. Finally, the technique for encoding, generating an initial population from multiple seeds along with the pseudo code of multiple seeds based genetic algorithm is explained.

Chapter five deals with the tests and analysis of the experimental results. To show the effectiveness of the proposed algorithms, the experimental analysis of these approaches are discussed in this chapter. The performance of the GeneticMax algorithm for mining maximal frequent item sets is shown through the experimental results. The experimental results of the Hybrid GeneticMax algorithm, along with the comparative analysis of this method with GeneticMax algorithm is demonstrated in this chapter. The experiments of the PSO based approach for mining association rules, for both frequent and infrequent

items is evaluated. The performance analysis of ARMGAAM and MBAREA algorithms for mining Boolean association rules are carried out on different real world data sets. Finally, the experimental results of multiple seeds based genetic algorithm is analyzed.

Chapter six furnishes conclusions and further research directions. This chapter summarizes how the research work presented in this thesis has obtained its stated goals. Further research directions that are identified during the research work and closing remarks about the effectiveness of evolutionary algorithms for mining interesting association rules, are also discussed in this chapter.

Chapter seven represents references that are studied for conducting this research.

1.6 Benchmark Data sets

Throughout this thesis, the proposed approaches are tested on different data sets for mining maximal frequent item sets and Boolean association rules. These data sets are carefully chosen from the University of California at Irvine (UCI) machine learning repository (http://archive.ics.uci.edu/ml/datasets.html) and University of Regina (http://www2.cs.uregina.ca/~dbd/cs831/notes/itemsets/datasets.php). The data sets are selected to have different numbers of attributes (from 8 to 118) and instances (from 101 to 12960), and different attribute characteristics (Boolean, categorical). These data sets are used as representative samples on which the proposed algorithms are evaluated to solve the research problems.

Chapter 2 - Literature Review

2.1 Introduction

This chapter provides a review on the literature that supports the motivation and forms the background of the thesis. Along with the intersection of data mining techniques and genetic algorithm domains, different areas within those fields such as mining frequent patterns, extracting association rules, multi-objective evolutionary algorithms are discussed through the following sections. This chapter concentrates on theoretical and technical background of the above mentioned fields and describes the underlying concepts with examples, when necessary. Section 2.2 covers the essential background of data mining techniques. This section reviews the typical related works in mining frequent patterns and association rules using conventional approaches and evolutionary methods. The basic concepts of genetic algorithm, interestingness measures and the essential background of multi-objective evolutionary approaches are covered in section 2.3. Finally, the technical background and related works of an initial population in genetic algorithm is discussed in section 2.4.

2.2 Data Mining

Data mining is the process of finding relationships in large data sets applicable to methods of artificial intelligence, statistics and machine learning. Different activities are included by data mining. It considers data from different sources and then translates, formats and cleans these data sets for further use, such as analysis, integration and validation. The goal of data mining is to extract patterns and knowledge instead of mining data itself from large amounts of data sets. By analysing large amounts of data, data mining is used for extracting unknown important patterns such as association rule mining, anomaly detection, and clustering. Predictive modelling, clustering techniques, summarization methods, link analysis and classification techniques are the five analytical domains which demonstrate the importance of data mining in real world applications. In business transactions, frequent pattern mining gives an idea about the popularity of buying item sets to the users. By using this information industries stock those popular products and gain benefitted by it. Initially association rules were used in market-basket analysis, however its application has now been extended to different real world fields including e-commerce, telecommunication, intrusion detection, bioinformatics, web mining, etc. (Han & Kamber 2006, pp. 9-39).

In human genetics research, the aim of sequence mining is to finding the changes in DNA sequences of individuals which are responsible for increasing the risk of common diseases such as cancer (Cameron & Leung 2011; Mabroukeh & Ezeife 2010). Frequent pattern mining plays a vital role in mining correlations, associations and other interesting relationships among data sets (Han et al. 2007). Moreover, it helps in different data mining tasks such as indexing, clustering, classification and so on. For this reason, mining frequent patterns and association rules are important data mining tasks and focused topics in the data mining research area.

2.2.1 Preliminaries

In this section, some popular notations, idea and conceptual diagrams for mining frequent patterns are introduced. Initially, databases are described through bitmap and bipartite graphs. Finally, the maximal frequent items sets and subsets of items are represented graphically using a lexicographic tree.

2.2.1.1 The Idea of Fast Response

If a data tuple contains long item sets, it generates huge candidate item sets which finally reduces the efficiency of a solution. A long item set enumerates combinatorial number of shorter, frequent sub item sets (Gouda & Zaki 2005). For example, a data tuple contains 50 item sets, such as $\{i_1, i_2, i_3, \ldots, i_{50}\}$ which enumerate $\binom{50}{1}$ frequent 1-itemsets: $(i_1, i_2, \ldots, i_{50})$, $\binom{50}{2}$ frequent 2-itemsets:(i_1, i_2), (i_1, i_3), … , (i_1, i_{50}), (i_2, i_3), (i_2, i_4),…, (i_2, i_{50}) and so on.

Lemma 1: If the length of an item set is n, then it enumerates $2^n - 1$ frequent sub-itemsets.

This sequence is too huge for a computer to compute and store if the length of an item set is long. For each sub-item set, Apriori algorithm (Agrawal & Srikant 1994; Hipp et al. 2000) needs to be used to scan the database and calculate the support value of that item set which increases the computational time of the algorithm and decreases the efficiency of it (Gouda & Zaki 2005).

An item set *I* is called maximal frequent itemset if the super item set of *I*, denoted by $\acute{I}$, is not frequent such that $I \subseteq \acute{I}$ (Dou et al. 2008; Salleb et al. 2002). Here $\acute{I}$ is an infrequent item set based on a support value defined by a user.

Lemma 2: If an item set *I* is a frequent itemset then all the subsets of *I* are frequent, based on a support value which is defined by a user.

For example, if an item set $I = \{1,2,3\}$ in set $S = \{1,2,3,4\}$ is frequent, i.e. $\sigma(I) \geq$ min_supp, then all the subsets of *I*, i.e. $\{1\}, \{2\}, \{3\}, \{1,2\}, \{1,3\}, \{2,3\}$ are frequent itemsets, based on a support value defined by a user (Agrawal & Srikant 1994). In Apriori algorithm, it scans the databases for all the subsets of *X* to get the support value. It takes huge computational time if the length of an item set is long. The computational time for mining frequent patterns is proportional to the length of an item set (D. Burdick et al. 2005). From the above discussion it can be concluded that, if a system is designed in such a way that if the generated chromosome is $I = \{1,2,3\}$ and it satisfies a user defined support value then it will not test all the subsets of *I* which dramatically reduces computational time for scanning the database.

2.2.1.2 Bipartite Graph and Bitmap Representation

If *U* and *V* are disjoint sets of vertices and *E* is the set of edges which connect the vertices *U* and *V*, then a bipartite graph is represented as a triple, i.e. $G = (U, V, E)$ where $E \subseteq U \times V$.

A binary matrix is a matrix of $m \times n$, where each entry consists of a value which is either 0 or 1 (Chen et al. 2006). Mapping between binary matrices and databases of transactions can be done in a straight forward way. Consider a database *D* which consists of transactions $\{t_1, t_2, \dots, t_{m-1}, t_m\}$ corresponding to rows and items $\{i_1, i_2, \dots, i_{n-1}, i_n\}$ corresponding to columns. The database *D* is a $m \times n$ matrix, where each entry is defined as a_{ij}. The value of a_{ij} is 1, if transaction t_i contains item i_j otherwise it is 0. Now each transaction is mapped as a set of items from the binary matrices.

Example 1: Consider a database *D* which consists of the following transactions t_1, t_2, t_3, t_4, t_5 and items i_1, i_2, i_3, i_4, where $t_1 = \{i_1, i_2, i_3\}$, $t_2 = \{i_1, i_2, i_3, i_4\}$, $t_3 = \{i_1, i_3, i_4\}$, $t_4 = \{i_1, i_3, i_4\}$ and $t_5 = \{i_1, i_2, i_3, i_4\}$. In database *D*, all the items are different.

Figures 1 and 2 show the bipartite graph (Zaki & Ogihara 1998) and the binary matrix of the database *D*:

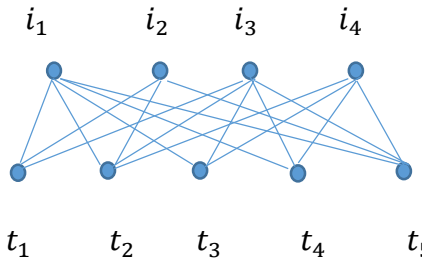


Figure 1: Bipartite graph representation of the database D

	i_1	i_2	i_3	i_3
t_1	1	1	1	0
t_2	1	1	1	1
t_3	1	0	1	1
t_4	1	0	1	1
t_5	1	1	1	1

Figure 2: Binary matrix representation of the database D

From Figure 2, if each transaction is mapped by items then the transactions are as follows:

$t_1 = \{1,1,1,0\} = 1110$

$t_2 = \{1,1,1,1\} = 1111$

$t_3 = \{1,0,1,1\} = 1011$

$t_4 = \{1,0,1,1\} = 1011$

$t_5 = \{1,1,1,1\} = 1111$

2.2.1.3 Maximal Frequent Item Sets and Lexicographic Tree

Item set *I* consists of n items, i.e. $I = \{i_1, i_2, i_3, \ldots, i_n\}$. X_k represents an item set containing *k*-items, where *k* = 1,2,…,n and $X_k \subseteq I$. If *k*=1, then X_k contains a 1-item, i.e., $X_1 = \{i_1\}$. If *k*=2, then $X_k$contains 2-items, i.e., $X_2 = \{i_3, i_4\}$, and so on. An itemset is called frequent if its support value satisfies a user defined support value and it is denoted FI. An item set *X* is called maximal frequent item set if it is frequent and no superset of *X* satisfies any user defined support value (denoted by MFI) (Doug Burdick et al. 2005; Wang et al. 2003).

Some research studies consider a search space which consists of all feasible solutions. A Lexicographic tree (D. Burdick et al. 2005; Huang et al. 2004) is the search space for searching maximal frequent item sets. A Lexicographic tree maintains lexicographic

ordering of items *I* in a database *D*. If an item *i* occurs before an item *j* in a database *D*, then it maintains lexicographic ordering, i.e., $i \leq_L j$. If there are two subsets S_1 and S_2, where $S_1 \subseteq S_2 and\ S_1, S_2 \in S$, then it maintains the following lexicographic order: $S_1 \leq_L S_2$. There is no lexicographic ordering relationship between two subsets $S_1$and S_2, if $S_1$and S_2 are disjoint subsets.

Figure 3 shows an example of a lexicographic tree which considers lexicographic ordering for four items. The root of the tree is an empty set and each *k*-level contains *k*-items. In each level, *k*-item sets maintain lexicographic ordering with the tail nodes containing items lexicographically larger than elements of the head node. The support value of the head node is more than that of the tail node. It can be seen that the nodes closer to the root are more frequent than those far from the root. There is a non-linear line (called a cut) in the tree which separates frequent item sets from infrequent ones. The nodes which are above the cut are frequent item sets and the elements below this cut are infrequent ones (D. Burdick et al. 2005).

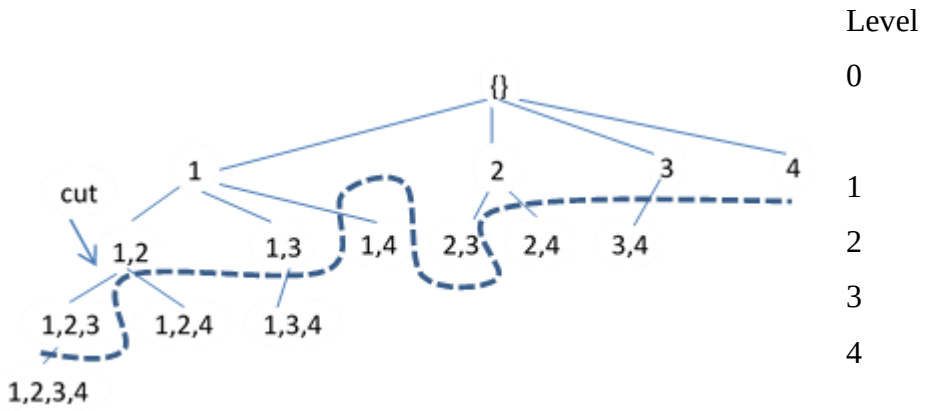


Figure 3: Lexicographic tree of four items

2.2.2 Related Works for Mining Frequent Patterns

In data mining research, frequent pattern mining is one of the challenging and focused areas for over a decade (Cameron & Leung 2011). A large number of literature and research works have been consecrated to this research area and extraordinary progresses have been made because of this dedicated effort (Kantardzic 2003, pp. 329-354; Han & Kamber 2006, pp. 227-248). Progresses happened in sequential pattern mining, correlation and structured pattern mining, scalable and efficient algorithms are designed for mining frequent item sets and so on. The scope of data analysis has been expanded by

the research of frequent pattern mining and will have profound effect on the methodologies and applications of data mining for further exploration. Though there are lots of progresses happened in frequent pattern mining research but still there are some challenging issues in this research area that need to be resolved (Han et al. 2007). The critical research issues which need to be considered by the future researchers are as follows:

Scalable mining methods are extensively studied, which are the focused topics in frequent pattern mining research (Han et al. 2007; Han & Kamber 2006, pp. 234-248). Current mining methods are used to derive sets of frequent patterns (Hipp et al. 2000; Agrawal & Srikant 1994; Borgelt 2012; Borgelt 2003). These sets of frequent patterns are too huge to use effectively. To reduce these huge sets, researchers have proposed several methods such as maximal patterns, representative patterns, closed patterns, condensed patterns and so on (Han et al. 2007; D. Burdick et al. 2005; Wang et al. 2003). But it is still undefined for a specific application which pattern set provides compactness and the quality of representation. Much investigation is needed to reduce the size of derived set of patterns and to increase the quality of preserved patterns (Han et al. 2007).

Some applications prefer approximate frequent patterns (Zaki 2001) although current studies show that efficient methods are available for mining complete and explicit complete set of frequent patterns. In bioinformatics to match with the biological entities one could be interested in searching long sequence patterns in DNA analysis (Agrawal & Srikant 1995; Ykhlef & ElGibreen 2009). Much investigation is needed to design efficient methods to make this mining more competent than the present tools available in bioinformatics.

Classification is another significant task in data mining research areas (Thabtah 2007). In data mining, classifications using frequent patterns means which frequent patterns are more adequate over another (Liu et al. 1998). In the future researcher should design a method in such a way that effective frequent patterns are mined directly from data (Han et al. 2007).

Some applications need profound understanding of patterns and interpretation of those patterns (Bayardo 1998; Jesus et al. 2011; Borgelt 2012; Agrawal et al. 1993). Most of the researchers have focused on discovering frequent patterns but have given less attention to analyzing and interpreting those patterns (Han et al. 2007). The semantic analysis of a pattern includes the meaning of that pattern, the typical transactions that pattern con-

siders and so on. The reason behind the frequency of a specific pattern is termed as a contextual analysis of a frequent pattern. For example, a pattern could be frequent depending on specific time duration, location, weather and so on. To improve the interpretability, effectiveness and usability of a frequent pattern, it is necessary to have deep understanding of frequent patterns (Mei et al. 2006).

It is well known that Apriori algorithm (Hipp et al. 2000) generates a candidate set and tests it in a breadth first manner. It discovers all the frequent item sets at level k before moving to its next level (k+1). It counts the support value of each node in level k and prunes those nodes if the support values of those nodes do not satisfy a user define support value. It generates candidate item sets at each level and scans the data sets so frequently that it is costly, especially when there exists a long pattern (Agrawal & Srikant 1994).

Pincer-Search algorithm (Lin & Kedem 2002) traverses a lattice through a bi-directional method that follows both top-down and bottom-up approaches. To find a maximal frequent item set it applies pruning methods by the following two properties:

1) All the subsets of frequent item sets are pruned

2) All the supersets of infrequent item sets are pruned.

Breadth first traversal (a level by level search strategy on a search space) is applied for a MaxMiner search algorithm. To prune the branches of a tree it performs a look-ahead method. MaxMiner uses breadth first approach for limiting the number of passes over the data sets but look-ahead, which involves superset pruning, works better for depth first search methods (Bayardo 1998).

DepthProject performs depth first traversal on a lexicographic tree along with variations of superset pruning. To order child nodes, it applies dynamic reordering methods. By trimming infrequent items out of each node’s tail, it reduces the size of the search space. To eliminate non-maximal frequent item sets DepthProject would require post pruning methods (Agarwal et al. 2000).

MAFIA, proposed by Burdick, Calimlim, and Gehrke (D. Burdick et al. 2005), extends the idea of DepthProject. Similar to DepthPorject, MAFIA also uses vertical bitmap representation where the support value/count of an item set is based on AND operations

among the item sets. For example, if there are 4 items in a data tuple and the data sets are as follows:

A	B	C	D
1	0	1	1
0	1	1	1
1	0	1	0
1	0	1	1
1	1	0	1

Figure 4: Vertical bitmap representation

Bitvectors for item set A,B,C,D of Figure 4 are 10111, 01001, 11110, 11011 respectively. To get the support value/count of the item set it needs to apply bitwise AND (&) operation between the bitvectors of the item sets. For the above example, the result of bitwise AND operation of bitvectors of items A and C is 10111 & 11110, which equals to 10110. The support value or count of an item is the number of 1's in the bitvector. Here the support value of an item set {A, C} is 3. If another bitvector D is added with the previous result of bitwise AND operation of bitmap {A,C}, it equals to 10110 & 11011, which equals to 10010. The support value of the item set {A,C,D} is 2. The search strategy of MAFIA integrates depth first method to traverse the tree to find maximal frequent item sets along with effective pruning methodology. Look-ahead pruning methodology which was first used by MaxMiner is also used by MAFIA. The last checking method of MAFIA is easy to test. Without counting, it allows us to conclude that {A,C} is frequent. This technique is defined as Parent Equivalence Pruning in.

In (Gouda & Zaki 2005), Gouda and Zaki proposed a novel approach called GENMAX to find maximal item sets. In this approach they used a novel technique called Progressive Focusing. This technique maintains local maximal frequent item sets (LMFI) which is used for making comparison with newly found frequent item sets (FI). Non maximal frequent item sets are identified through this step and it decreases the number of subset testing. GENMAX uses vertical representation of a data set and stores transaction identifier set (TIS) for each item set instead of bitvector. The support value of an item set is defined by the cardinality of an item set's TIS. Researchers of GENMAX concluded that, through experimental results this algorithm performs better than existing algorithms on different types of data sets.

Bilal Alataş and Erhan Akin (Alataş & Akin 2005) designed an efficient genetic algorithm as a search strategy to mine both positive and negative quantitative association rules. Association rules are deduced from frequent patterns. Their approach is different from other methods. This method mined the association rules without generating frequent item sets. The proposed genetic algorithm does not depend on minimum support and confidence value which is hard to define for a data set. A new genetic operator named uniform operator is used in this approach which ensures genetic diversity.

Another interesting problem in data mining is classification (Thabtah 2007; Wang et al. 2007). Different lengths of item sets are classified into different groups based on the frequency of the item sets. Concise symbolic rules with higher accuracy are mined from neural network. To get the required accuracy, network is initially trained. Network pruning algorithm is used to prune redundant connections. Classification rules are generated through the result of the analysis of activation values of the hidden layers. Researchers noticed main drawback of using neural network in different data mining test problems is the training time. Though it provides lower classification error rate than decision trees but it requires a long training time (Lu et al. 1996; Ghosh et al. 2011).

In (Dou et al. 2008), quick response data mining model based on a genetic algorithm has been designed. This approach gives more flexibility to the user. Long frequent item sets are generated because of the higher relationship among data tuples. If Apriori algorithm mines frequent item sets from these tuples it could take a huge amount of time because it needs to access the data sets so frequently and large number of candidate item sets are generated. This approach avoids considering huge candidate item sets. It only scans the data sets for those frequent item sets users are more interested. This system uses a genetic algorithm to mine item sets and then it show it to the users. If the users are interested then it scans the data sets for the support value of those item sets.

Hipp, Guntzer and Nakhaeizadeh (Hipp et al. 2000) showed the performance analysis of Apriori and other existing famous algorithms of present day. Although Apriori algorithm was invented long time ago but still it is one of the famous algorithm and performs better than other existing algorithms like Eclat, Partition, DIC and so on for large value of min_supp. On the other hand, other algorithms perform better than Apriori for small value of min_supp. Finally, they concluded that no algorithms fundamentally beating each

other. After analyzing, they (Hipp et al. 2000) showed that the run time behavior of all the algorithms are similar as it is expected.

2.2.3 Previous Studies for Mining Association Rules

It is well know that the association rule mining task is first introduced by Agrawal et al. (Agrawal et al. 1993). They develop Apriori algorithm which generates a candidate set and tests it in a breadth first manner. Before moving to its next level (k+1), this approach discovers all the frequent item sets at level k. The support value of each node is calculated at level k and prunes those nodes if the support values of those nodes do not satisfy a user define support value. It generates candidate item sets at each level and scans the database so frequently that it is costly, especially when there is a long pattern and large transaction database. After generating frequent item sets, the next step is to generate rules form those frequent patterns. A rule is valid if it satisfies a user defined confidence value. Confidence value of a rule $X \rightarrow Y$ is defined by $\frac{supp(X \cup Y)}{supp(X)}, where\ X, Y \subseteq I, supp(I) \geq \min\_supp$. Apriori algorithm follows support-confidence framework, introduces the following major issues:

1) This algorithm highly depends on user define support and confidence values, although users have no knowledge regarding the database.
2) If the support value is too high, generation of frequent item sets is less and hence few rules are mined. If the support value is too low, then almost possible patterns will become frequent and a large number of unnecessary rules are generated.
3) Since this algorithm only use a single criteria i.e. confidence to evaluate the quality of a generated rule, it generates misleading rules as well (Zhou & Yau 2007).

Some recent studies focused on designing different efficient association rule mining algorithms where the first task is to find frequent item sets (Webb 2000; Toivonen 1996; Zhou & Yau 2007; Wu et al. 2002; Wu et al. 2004). The main limitation of these approaches need multiple passes over the transaction database to find frequent item sets. A large number of disk I/O is required for a large database since database are disk resident and for each pass it needs to read the database completely. The number of disk I/O depends on the size of the database (Yan et al. 2005).

The diversity, application, usefulness and probability of genetic algorithm for mining high utility item sets containing negative item values for transaction database is showed by Kannimuthu and Premalatha (Kannimuthu & Premalatha 2014). Mutation operator is used to maintain diversity from one generation of population to the next one. Generally, the mutation probability is set to low. The searching task is becoming primitive random search if the mutation probability is set too high. To avoid this, instead of using fixed mutation rate they use a ranked mutation approach because it gives better result (Premalatha & Natarajan 2009). Initially a large mutation rate is applied for exploring more on the search space. The ranking of offspring depends on the fitness value. The mutation rate is assigned for an offspring based on the rank of that offspring. The rate of mutation is set low for a higher ranked offspring and through this way the offspring containing highest fitness value may reach to the optimal solution in high possibility.

To mine quantitative association rules researchers propose a new algorithm which is based on genetic algorithm named QUANTMINER (Salleb-aouissi et al. 2013; Salleb-aouissi et al. 2007). By optimizing support and confidence value, this system dynamically identify good intervals in association rules. Researchers apply this algorithm in different data sets and showed the usefulness of this algorithm as a data mining tool.

R.J.kuo and C.W.Shih use a new meta-heuristic technique name ant colony system (Kuo & Shih 2007) to mine large database for efficient searching of association rules. Multi-dimensional constraints are considered in this approach. In addition this approach also considers user's assign constraint. The experimental result shows that it gives more condensed rules than Apriori algorithm. The computational time of this approach is less than Apriori algorithm. Although this system provides promising results but this system still faces some issues which need to resolve. After analyzing the results it found that lots of similar rules are generated so the researchers suggest another technique like fuzzy approach to merge those similar rules into one class.

Researchers propose a novel particle swarm optimization algorithm, named rough particle swarm optimization algorithm (RPSOA) to mine numeric association rules. Based on notion of rough patterns, this algorithm uses rough decision variables and rough particles. As opposed to precise values, a rough particle consists of upper and lower values. Conventional and rough particles and variables are used by this proposed method. This approach is designed in such a way that it simultaneously search the intervals of numeric

attributes and extract the numeric association rules based on these intervals. Furthermore, this approach directly mines association rules from a data set without generating frequent item sets. Since there was no study for mining association rules, researchers conclude that this approach gives satisfactory results in its first application (Alatas & Akin 2008b).

Most of the association rule mining tasks assume that items in a dataset have a uniform distribution. By weighting individual items, weighted association rule mining tasks are used to provide a notion of importance to an individual item. To assign meaningful weights to each item for mining association rules, researchers introduces a new approach named Weighted Association Rule Mining using Particle Swarm Optimization (WARM SWARM). In this proposed approach researchers use PSO to search the vector space of possible solutions for finding the optimal solution of a problem. Unlike Apriori, this approach multiplies the support of an item by the sum of the weights of its constituent items during the generation of candidate item sets (Pears & Koh 2012).

To mine association rules most researchers focus on ameliorating computational efficiency. To determine the threshold values of support and confidence which are the key factors for association rule mining task, researchers propose a new approach which is based on a particle swarm optimization technique. Suitable fitness values and their corresponding support and confidence values of identified swarms are searched through this approach. Their result show that particle swarm optimization algorithm quickly finds suitable threshold fitness values of item sets and quality rules are obtained through this way. Users can mine specific rules from a large database by setting support or fitness values. Since this technique free from support constraint, the main problem of this approach is users have no control over mining techniques. Apart from this their result only shows two or three dimensional rules instead of more dimensional rules which could be interesting for the policy makers of the industry (Kuo et al. 2011).

2.3 Genetic Algorithm

Genetic algorithm considers adaptive methods which are used to solve search as well as optimization problems. This algorithm is inspired by natural selection and the “survival of the fittest” mechanisms which are clearly stated by Charles Darwin in the book name “The Origin of Species”. Based on the fitness value, in a competing environment only the stronger individuals will survive. The processes in natural population which are essential for evolution are simulated by GA. Holland (Man et al. 1996) first proposed the

basic principles of genetic algorithm. Thereafter, a large number of researchers worked on genetic algorithm (Beasley et al. 1993; Srinivas & Patnaik 1994; Michalewicz 1992, pp. 50-91). Naturally individuals are competing with each other for their shelter, food, clothes, water and so on. Even members of the same class often compete to attract their partner. Those individuals are referred to as strong if they are successful in surviving and attracting a partner. A large number of offspring is produced by strong individuals. On the other hand poorly performing individuals are referred to as weak and have less probability to produce newer offspring. The combination of good attributes from different parents can produce "superfit" offspring. That is the fitness of this offspring is higher than the fitness of the parents. In this fashion, species becoming more and better suited in the present environment.

Genetic algorithm plays a vital role for this study which simulates the natural behaviour of biological organisms. Genetic algorithm based techniques are robust and can be used to solve a wide range of problems including those which are hard to solve by other methods. Researchers conclude that, it is not guaranteed that GA always provides optimum solutions to a problem rather it provides "acceptably good" solutions to a problem which is solved by other methods "quickly" (Beasley et al. 1993). Existing methods which are working well as a solution for a particular problem, improvement of those methods can be done by hybridizing with GA (Wan & Birch 2013).

A Traditional Genetic Algorithm generates an initial population, and then computes the fitness value of that population. Two individuals are selected from the old generation and applying crossover, mutation operators to produce two offspring. It selects the survivors which have the best fitness value and inserts those in the new generation. If the population is converged to a solution then the algorithm is terminated. In this algorithm, fitness function provides the fitness value of an offspring which is a specification of the offspring (Man et al. 1996; Beasley et al. 1993).

2.3.1 Development of a New Mutation Operator

Several researchers introduce a new mutation operator to improve the performance of a Genetic Algorithm. They applied this technique in the well-known application named Traveling Salesman Problem (TSP) for finding shortest distances (Helsgaun 2000). This method is referred to as a Greedy Sub Tour Mutation method (GSTM).

Traveling Salesman Problem is one of the important combinatorial optimization problems. This problem is based on finding shortest paths among “n” cities. A salesman follows the shortest path to visit all the cities once and comes back to the starting city. The Performance of travelling salesman problem is examined with respect to solution time and error value. Previous researchers have considered approximating solutions which provide low error values and quick generation of a solution (Jayalakshmi et al. 2001; Seo & Moon 2002; Stutzle & Hoos 1997). Although approximating algorithms may give good solutions but they do not guarantee optimal solutions. Approximate algorithms are simple to design with very short run times. If an application needs a solution which deviates only a few percentages from the exact solution then approximate algorithms are an appropriate choice (Helsgaun 2000).

A Genetic Algorithm which is developed by Holland in 1975 (Man et al. 1996), is a random search algorithm by generating population iteratively. This algorithm is used to find approximate solutions for further optimization (Anon 2014). The main aim of using a genetic algorithm is to reach good results by discarding bad solutions during generation of populations (Freitas 2003). The basic steps of genetic algorithm are as follows:

Procedures of Genetic Algorithm

Step 1: Generate an initial population

Step 2: Find the fitness value of that population.

Step 3: Select parents for reproduction.

Step 4: Generate new chromosomes by applying selection, crossover and mutation.

Step 5: Go to Step 2 with newly generated chromosomes until termination condition is reached.

Researchers have introduced different mutation operators. For example, insertion mutation (Fogel 1988), exchange mutation (Banzhaf 1990), inversion mutation (Fogel 1993), simple inversion mutation (Grefenstette et al. 1985) and so on. The reason of using mutation operator in genetic algorithm is to elude local solutions (Albayrak & Allahverdi 2011). During optimization it is possible to avoid local solutions by using mutation operators. These mutation operators have no impact on developing shortest paths. The operators which are used to develop a tour are called greedy methods. The drawback of

greedy methods is that when it reaches local solutions it is stuck down with that solution and hard to jump in any other unprecedented solutions. Therefore the result of a greedy method does not obtain optimal solution.

P_{RC}: Reconnection Probability
P_{CP}: Correction of Perturbation Probability
PL: Linearity Probability
L_{MIN}: Minimum Sub Tour Length
L_{MAX}: Maximum Sub Tour Length
NL_{MAX}: Neighbourhood List Size
Step 1: Define randomly starting (R_1) and ending points (R_2) of sub tour;
Step 2: According to the random (R_{nd}) number produced between (0-1);
If ($R_{nd} \leq P_{RC}$) then,
{
Subtract [R_1-R_2] sub tour from tour T($T^{\#} \leftarrow$T-[R_1-R_2];$T^{*} \leftarrow$[R_1-R_2]);
Hold on T^{*} sub tour that causes minimum extension to T^{*} tour;
} Else{
According to the random (R_{nd}) number produced between (0-1);
If ($R_{nd} \leq P_{CP}$) then,
{
Copy [R_1-R_2] sub tour in the T tour ($T^{*} \leftarrow$[R_1-R_2]);
Add each element of T^{*} sub tour to the T tour starting from the position R_1 by rolling or mixing with P_L probability;
} Else {
Select randomly one neighbour from neighbour lists for the points R_1 and R_2(NL_{R1} and NL_{R2});
Invert the points NLR1 or NLR2 that provides maximum gain such a way that these points will be neighbours to the points R_1or R_2;
Repeat if the inversion is not taken place;
}
}

Figure 5: Algorithm of greedy sub tour mutation

Greedy sub tour mutation operator includes both the classical and greedy techniques to obtain the optimal solutions for Travelling Salesman Problem. Since this operator uses a parametrical structure, it does not stick down with local solutions and jumps in any other unprecedented solutions. Figure 5 (Albayrak & Allahverdi 2011) describes the Greedy Sub Tour mutation algorithm.

2.3.2 Techniques for Improving Genetic Algorithm in an Application

In pattern recognition research, selecting a feature is an important research part. The objective of this application is to select an optimal subset from a large primary feature set (Lin et al. 2008). The term optimal subset means that, by using this type of feature subset it is possible to detect and recognize the best target efficiently with low computational cost. To design pattern classifiers, feature selection considers three goals: 1) reducing the cost of feature extraction, 2) improving the accuracy of classification and 3) improving the reliability of the estimation of performance (Kudo & Sklansky 2000).

An algorithm is used to select the feature and the key components of this algorithm is an evaluation function and a search methodology. Search strategy is used to find the best combination of features (Uncu & Türkşen 2007) and it is classified into three groups: exponential, sequential and stochastic. Exponential search expressed as exhaustive and branch and bound methods. Exhaustive search is impractical when it is used to search optimal feature subset in high dimensional space due to its massive computational cost. By pruning branches of search tree, branch and bound method reduces the search time. The main drawback of this method is that it requires monotonic evaluation function which is sometime false (Chen 2003). Sequential forward search, sequential backward search and plus-l take-away-r are including by sequential search methods (Gunal et al. 2009). In sequential forward search when a feature is choose, it is not remove although it is redundant. This is the main drawback of sequential forward search method. Because of starting the searching with high dimensional feature space, sequential backward search costs massive computational time than sequential forward search. Another drawback of this method is that it cannot reselect the feature although it is useful for future purpose. The combination of sequential forward search and sequential backward search id are called plus-l take-away-r. This method is still costly and difficult to choose l and r of plus-l take-away-r method.

Because of the above drawback of the mentioned search methods, researchers become interested in an evolutionary algorithm named Genetic Algorithm. Genetic Algorithm is heuristic which mimics natural selection approaches to find an optimal subset in an efficient way. This algorithm performs global search and its time complexity is less than that of other algorithms (Ghosh et al. 2012), for the reason that Genetic Algorithm is based on a greedy approach. Researchers apply genetic algorithm on large range of optimization problems (Avci et al. 2009; Bhanu & Lin 2003; Song et al. 2009; Cho et al. 2008). A simple genetic algorithm includes: encoding of the chromosome, initialization of the population, calculating the fitness value of examined chromosomes, selection, crossover, mutation and termination condition. Multi-character feature set consists of different group of features with different characters. To get better classification accuracy, searching the optimal feature subset from multi-character feature set is the main objective in pattern recognition. This goal is not achieved through a simple genetic algorithm.

After applying selection, crossover and mutation genetic operators the newly generated chromosomes are sometimes similar to the previous ones. These replicated chromosomes are considered as invalid chromosomes since they have been tested before. It takes time to identify these invalid chromosomes. Researchers develop this approach further in such a way that it will avoid generating invalid chromosomes and increase the generation of more valid chromosomes, which will increase its speed in converging to a solution (Yang et al. 2011). An improved genetic algorithm would:

1) Manage the segmented chromosomes
2) Use the segmented crossover operators
3) Use the segmented mutation operators
4) For crossover and mutation, dynamically adjust the probability with respect to the number of generations and fitness values of population.

This technique introduce segmented crossover operator. Crossover plays a vital role in genetic algorithm and makes a distinct difference with other optimization algorithms. Generally used crossover operators are single, double, multipoint crossover and so on (Kaya 2009).

Following figure shows the mechanism of an improved genetic algorithm.

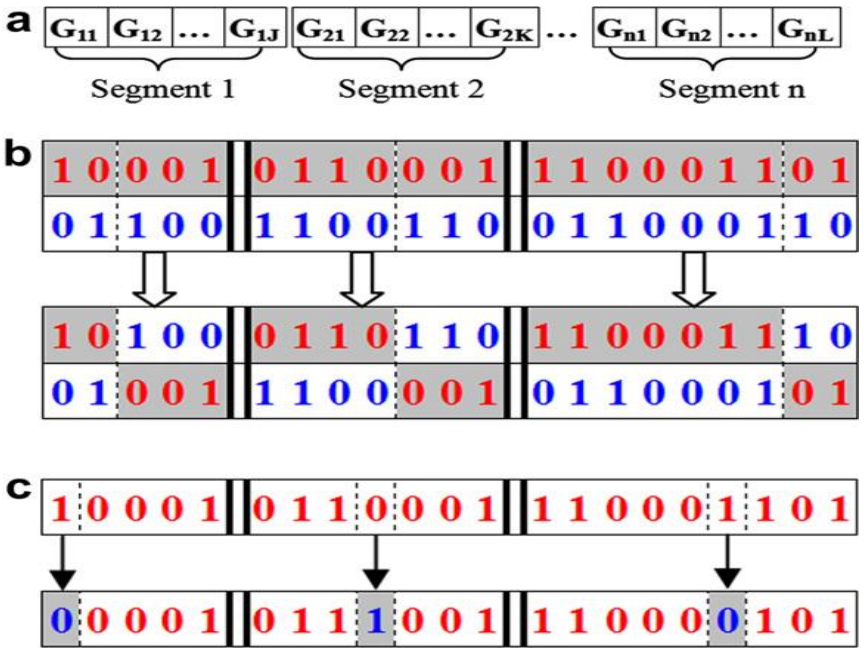


Figure 6: Segmented chromosome a) encoding b) crossover and c) mutation

Researchers have improved genetic algorithm when selecting optimal feature subset from multi-character feature set. Simulation results show that optimal feature subset selection is more efficient and effective when compared with multi-character feature set, using the improved genetic algorithm (Yang et al. 2011).

2.3.3 Interestingness Measures

To measure the quality of a rule, users could use more complex formulas to determine whether a rule is interesting or not (Martin et al. 2014; Geng & Hamilton 2006; Liu et al. 2000). Many interestingness measures have been proposed for different data sets for reducing the number of mined rules. These measures have been defined as a broad concept which encompass several features such as conciseness, peculiarity, surprisingness, generality, reliability, usefulness and so on (Geng & Hamilton 2006).

Let $I = \{i_1, i_2, \ldots, i_r, \ldots, i_{n-1}, i_n\}$ be an item set which contains n-numbers of items and the transaction data set be $T = \{t_1, t_2, \ldots, t_{k-1}, t_k\}$ which contains k-numbers of transactions. Each transaction of *T*, i.e. t_i where i ϵk, is a subset of an item set *I* is such that $t_i \subseteq I$. If A and B are two item sets, then an association rule between these item sets is defined as $A \rightarrow B$, where A is antecedent and B is its consequent and $A \subseteq I$, $B \subseteq I$, $A \cap B = \varphi$. Support and confidence are the two quality measurement factors for evaluating the validity of an association rule $A \rightarrow B$, which are defined as follows: the support and confi-

dence values of a rule A→B is defined by the term, *supp*(A∪B)= |(A∪B)|/|T| and *conf*(A→B) = *supp*(A∪B)/*supp*(A), respectively, the total number of records in a data set is defined by the term |T|. That is, support means the occurring frequency of an item set in a data set and strength of a rule is measured by confidence. A rule A→B is valid if *supp*(A∪B) ≥ min_supp and *conf*(A→B) ≥ min_supp, where min_supp and min_conf are an user defined support and confidence value (Agrawal & Srikant 1994; Hipp et al. 2000). However, several researchers have noted that support-confidence framework has led to the generation of a huge number of misleading rules. A rule A→B is misleading, if supp(B) > confidence(A→B) i.e. there is a negative correlation between the item sets of antecedent and consequent. High support based item sets are the source of misleading rules, since they exist in most of the records and therefore any items may seem to be a good predictor because of the presence of the high support based item sets. On the other hand, confidence measure does not take into account the consequent part of a rule. For this reason it does not identify negative dependence or statistical independence between item sets (Martin et al. 2014).

In recent years, several authors have proposed different measures according to the potential interest of the users (Geng & Hamilton 2006; Martin et al. 2014). Some of those that are used in the current literature for mining BARs are briefly explained.

The conditional probability (Yan et al. 2005; Yan et al. 2009) measure of a rule analyses the dependence between *A* and *B* and it is defined as,

$$CP(A|B) = \{supp(A \cup B) - supp(A)supp(B)\} / \{supp(A)(1 - supp(B))\} \quad (1)$$

Its obtain values in [- ∞, ∞], where misleading rules are represented by 0 > value > -∞, 0 < value < ∞ represents positive association rules, and value = 0/–∞/∞ represents trivial rules. The ratio between the confidence and the expected confidence of a rule is measured by lift (Ramaswamy et al. 1998) and it is defined as,

$$lift(A \rightarrow B) = supp(A \cup B) / \{ supp(A)supp(B)\} \quad (2)$$

The netconf (Ahn & Kim 2004) measure is used to evaluate a rule based on the support value of that rule and its consequent and antecedent support. Its domain range is [-1,1],

where positive values, negative values and zero represent positive dependence, negative dependence and independence, respectively. Netconf of a rule $A \rightarrow B$ is defined as,

$$\text{netconf}(A \rightarrow B) = [\text{supp}(A \cup B) - \{\text{supp}(A)\text{supp}(B)\}] / [\text{supp}(A)(1 - \text{supp}(A))] \quad (3)$$

For finding interesting rules, new rules are generated based on each item present in the consequent part of a rule. Since a number of items are present in the consequent part of a rule and it is not predefined, this approach may not be suitable for an association rule mining task. Recall the definition of interesting (Wakabi-Waiswa & Baryamureeba 2008), a new expression for measuring the interestingness of a rule $A \rightarrow B$ is defined as,

$$I = [\text{supp}(A \cup B)/\text{supp}(A)] \times [\text{supp}(A \cup B)/\text{supp}(B)] \times [\text{supp}(A \cup B)/|D|] \quad (4)$$

Here I is the interestingness constraint of a rule $A \rightarrow B$ and the total number of records in a database is defined by the term $|D|$. Its domain range is $[0, \infty]$, where 0, ∞ and $0 <$ value $< \infty$ represents independence, trivial rules and positive dependence, respectively.

2.3.4 Multi-Objective Evolutionary Algorithms for Association Rule Mining

An association rule is an implication between two item sets A and B, A→B, which is used to define the dependencies between the item sets in a data set. The problem of mining association rules are considered by many researchers and a large number of algorithms are developed for extracting association rules from different type of data sets (Hipp et al. 2000; Han & Kamber 2006, pp. 227-254).

Most of the existing classical algorithms for mining association rules are based on a support-confidence framework (Agrawal & Srikant 1994; Hipp et al. 2000; Jesus et al. 2011). This framework consists of two sub processes: finding all frequent item sets and generating rules from those frequent item sets based on a user defined support value and a confidence value, respectively. Several authors (Yan et al. 2009; Yan et al. 2005; Wakabi-Waiswa & Baryamureeba 2008; Jesus et al. 2011) have noted that these algorithms raised the following major challenges: 1) Users need to specify an appropriate threshold value for mining rules although they have no information regarding the data set, and 2) Association rule mining is an NP-Hard problem because searching all frequent item sets satisfying a minimum support value reveals an exponential search space

of size 2^n, where n is the number of item sets (Yan et al. 2009; Jesus et al. 2011). Finally, it generates a huge number of unnecessary rules from frequent item sets, resulting in weak mining performance (Berzal et al. 2002; Martin et al. 2014).

To avoid the use of minimum support and confidence threshold, researchers use genetic algorithm based multi-objective approaches because through this way, a more complex value is considered as a fitness function for an individual (Jesus et al. 2011).

Recently, a large number of research papers have used evolutionary algorithms for mining association rules. These studies have found that evolutionary algorithms (EAs) particularly genetic algorithms based approaches are efficient tools especially when the search space is too large to use deterministic search methods (Martin et al. 2014; Mukhopadhyay et al. 2014). Because of inherent parallel structure, GA based methods are effective for automatic processing of large amount of data and discovering meaningful and significant information. In real world applications, data sets not only use quantitative or numeric values but also contain categorical values. For this reason, several studies are proposed for mining Boolean association rules (BARs) from data sets with categorical values (Yan et al. 2009; Shenoy et al. 2003; Shenoy et al. 2005).

Ghosh and Nath (Ghosh & Nath 2004) consider the association rule mining task as a multi-objective problem instead of a single objective one. Different measures are used to improve the quality of a generated rule such as support count, comprehensibility and interestingness. Using these measures as an objective for association rule mining task, this study uses a pareto based genetic algorithm to mine useful and interesting rules from market basket database.

To mine interesting association rules, Wakabi-Waiswa and Baryamureeba (Wakabi-Waiswa & Baryamureeba 2008) proposes a Pareto based multi-objective evolutionary algorithm. For improving the interestingness of an association rule, they use different measures such as J-measure, perplexity, comprehensibility, interestingness and predictive accuracy.

Yan, Zhang and Zhang (Yan et al. 2005; Yan et al. 2009) proposes ARMGA and EARMGA algorithms for identifying BARs using genetic algorithm without specifying actual minimum support and confidence value. This article showed the hardness of selecting suitable threshold values by the users since different database require different

support values to mine useful and interesting rules. Instead of using support-confidence framework, these algorithms use Piatetsky-Shapiro (Piatetsky-Shapiro 1991) based rule interest method to define the positive confidence of a rule. To encode each association rules these algorithm follow Michigan strategy based encoding technique. Experimental results show that, a large number of high quality rules are generated due to considering weak fitness function. Because of the use of simple genetic operators like mutation, these approaches miss some high quality rules which are generated in intermediate generation of a population.

Recent multi-objective association rules with genetic algorithm (ARMMGA) is proposed for reducing the generation of a large number of rules by ARMGA (Qodmanan et al. 2011). New crossover and mutation operators are presented in this approach to prevent the generation of invalid chromosomes in ARMGA. In this approach, the order of the chromosomes in the population is specified by the fitness value. Although this approach generates a smaller number of rules but some of those are misleading and trivial due to using a weak constraint. The fitness function is defined in such a way that it generates unnecessary rules.

In order to extract a set of high quality rules which are easy to understand and interesting, recent studies show that researchers jointly optimize different measures (Alatas & Akin 2008a; Ghosh & Nath 2004; Martin et al. 2014). These approaches remove the drawbacks of single objective algorithms and mine high quality rules from the data sets with quantitative or numerical values (Salleb-aouissi et al. 2013; Webb 2001; Martin et al. 2014).

Motivated by the features of multi-objective approaches, in this thesis two new GA based approaches are proposed which are based on different design factors and data sets that jointly optimize multiple objectives for discovering a reduced set high quality BARs. The main objectives of designing these approaches are generating rules which are easy to understand, interesting and having a good trade-off among the number of rules, support, confidence and other objectives of the data sets.

2.4 Initial Populations of an Evolutionary Algorithm for Association Rule Mining Problems

An initial population has a significant effect of further generation of a population. Previous studies of a simple genetic algorithm which is based on a single seed, the effects of an initial population and dynamic diversity control mechanism in a genetic algorithm are described in this section.

2.4.1 A Single Seed Based Simple Genetic Algorithm

The main goal of a genetic algorithm is to achieve better solutions by discarding bad solutions during the generation of candidate populations from current to next generation (Albayrak & Allahverdi 2011; Srinivas & Patnaik 1994). The components of a simple genetic algorithm include encoding of chromosomes, initialization of a population based on a randomly selected single seed, calculating fitness value of individuals, selection, crossover, mutation, and stopping condition (Yang et al. 2011; Albayrak & Allahverdi 2011; Yan et al. 2009). The basic steps of a single seed based simple genetic algorithm is shown through the following flowchart:

PROCEDURE: **A SINGLE SEED BASED SIMPLE GENETIC ALGORITHM**	
Input D:	Data set D, S: Seed Chromosome, size_of_population, sp: Selection Probability, cp: Crossover Probability, mp: mutation probability
(0)	**Begin**
(1)	Generates an initial population P based on a seed chromosome S
(2)	**Repeat** step 2 to 6, until termination condition is satisfied
(3)	Calculate the fitness value of individuals in P
(4)	Select parents from population P for reproduction operation (o_a,$o_b \in P$) with a selection probability P_{sp}.
(5)	Reproduce new offspring by applying crossover operation on parent chromosomes with a crossover probability P_{cp}.
(6)	Applying mutation operation on new offspring with a mutation probability P_{mp}.
(7)	Replace new offspring with the previous one in the population
(8)	**End**

2.4.2 Effects of an Initial Population in Genetic Algorithm

Most of the association rule mining algorithms use single seed to initialize the population. For global optimization, GAs usually use a metaheuristic method, but relatively few research papers are published based on the techniques of the generation of an initial population. Traditionally, pseudo random chromosomes are used to generate an initial population. Current research shows that initial populations have significant effects of producing good solutions over several generations (Maaranen et al. 2007). Some researchers (Maaranen et al. 2004) use quasi random sequences to generate initial populations for GA. These sequences, which do not imitate random points are successfully used in computer simulations, quasi random searches and numerical integration (Snyder 2000).

The association rule mining algorithms which are based on GA, use a single seed chromosome for generating an initial set of solutions. These algorithms suffer from the following major challenges: 1) Different seed chromosomes generate different initial populations. Because of this reason different seed chromosomes yield different results. 2) It is a hard process to define a good seed for a specific application. 3) Defining seed is not an automatic process rather it is manual since the maximum range of a seed chromosome varies from data sets to data sets. For example, data sets A and B contain 100 and 50 items respectively. So, the range of a gene of a chromosome for a data set A, should be in between 1 to 100. On the other hand, for a data set B it should be 1 to 50.

To explore more search space and exploit it for further generation, a large number of research papers have been done, introducing new methods and genetic operators. As these methods and operators are problem specific, so the researchers (Chang et al. 2010) introduce an approach named dynamic diversity control in a genetic algorithm (DDCGA) for increasing the diversity in the chromosomes of a population. In this approach they maintain a proper balance between the exploration and exploitation search by regulating the diversity level of the generated population. If the diversity of the population drops down to the threshold level, then artificial chromosomes with high diversity are injected into the evolutionary process to increase the diversity of the population. The basic idea of this process is to generate the multiple archives by gathering high quality chromosomes from different initial seeds of a simple genetic algorithm. The system selects artificial chromosomes from multiple archives and inject these each

time the injection process is provoked. Through this technique, the diversity of the population is increased and the evolutionary process can explore more search space (Chang et al. 2010).

Motivated by the features of the diversity control approach, in this study a new genetic algorithm is proposed, named multiple seeds based genetic algorithm (MSGA), which is based on multiple seeds to initialize the population. To deal with the challenges raise by a single seed based approach, this proposed method subdivides the whole solution space into m-regions and randomly selects high quality seed chromosomes from each region. After selection of *m*-seeds from *m*-regions, this approach generates initial population from each seed using a Euclidean distance method. As a result, at the beginning of the evolutionary process MSGA obtains strong searching ability and generates an optimum result.

2.5 Summary

This chapter reviewed the main concepts of data mining techniques, genetic algorithm, mining frequent patterns, mining association rules, multi-objective optimization, and an initial population in genetic algorithm. This chapter also reviewed the related work of using classical methods and evolutionary algorithm based approaches for mining frequent patterns, association rules and single seed based genetic algorithm.

The limitation of the existing research works that lead to the motivation of this research were also discussed. This can be summarised as follows:

- Most of the existing association rule mining algorithms are based on a support-confidence framework. These algorithms suffer from the following major problems. Users need to specify an appropriate threshold value for mining rules although they have no information regarding the database. Association rule mining is an NP-Hard problem because searching all frequent item sets satisfying a minimum support value reveal an exponential search space of 2^n, where n is the number of item sets. Finally, it may generate a huge number of unnecessary rules from frequent item sets, resulting in weak mining performance. To avoid these problems researchers use genetic algorithm based approaches because a genetic algorithm is an efficient tool for a global search, especially when the search space is large enough to use deterministic search methods. Although the

existing genetic algorithm based approaches generate a smaller number of rules but some of those are misleading and trivial due to using weak constraints. The fitness function is defined in such a way that it generates unnecessary rules. In addition, because of the use of simple genetic operators like mutation, these approaches miss some high quality rules which are generated in intermediate generation of a population.

- Most of the association rule mining algorithms use a single seed for generating an initial population. Researches show that initial populations have significant effects of producing good solutions over several generations. However, single seed based evolutionary algorithms suffer from the following major challenges:
    1) Different seed chromosomes generate different initial populations. Because of this reason different seed chromosomes yield different results.
    2) It is a hard process to define a good seed for a specific application.
    3) Defining seed is not an automatic process rather it is manual since the maximum range of a seed chromosome varies from data set to data set.

Therefore, to avoid these problems, the following techniques are incorporated by the proposed approaches:

1) Developing mutation operators along with best population and re-initialization techniques to avoid the generation of iterative rules and keep the high quality rules which are generated in the intermediate generation of a population.
2) Strengthen the fitness function by using different measures such as minimum interest, lift, conditional probability and so on.
3) Subdivide the whole solution space into m-domain to get a seed from each domain. Finally, these seeds will be used to generate an effective initial population.

In the next chapter, the explanation of the main features and the theoretical aspects of the proposed methods will be described.

Chapter 3 - Research Methodologies

3.1 Introduction

The literature review chapter concluded with the limitation of the existing research works and summarized the research problems.

This chapter describes the proposed methods which are used to conduct this research to solve the research problems. At first, the requirements for developing the frequent pattern and association rule mining tasks are described in section 3.2 and 3.3, respectively. These sections are followed by introducing two new approaches in section 3.4, named GeneticMax and Hybrid GeneticMax for mining frequent patterns. Section 3.5 describes the method based on the particle swarm optimization technique for mining association rules. For discovering Boolean association rules from categorical data sets, new multi-objective evolutionary approaches such as adaptive and class based mutation methods are proposed in section 3.6. Finally, the architecture of multiple seeds based genetic algorithm is presented in section 3.7.

3.2 Requirements for Developing Frequent Pattern Mining Algorithm

There are five main requirements for developing an efficient maximal frequent item sets (MFI) mining algorithm. A set of techniques is needed which fulfils the following requirements:

1) It will not scan a database more than once for a specific item set.
2) If X is an item set in a positive boundary area and there are no supersets of X and it has already been tested, then all the subsets of X will be pruned and defined as invalid data sets.
3) If X is an item set in a negative boundary area and there are no subsets of X and it has already been tested, then all the supersets of X will be pruned and defined as invalid data sets.
4) It should maintain an interactive mining process, where users can change the threshold to get different sets of MFI.
5) It will give correct solutions for different sizes of databases.

Apriori algorithm and FP-Tree do not satisfy requirements 1, 2, 3 and 4 respectively (Han et al. 2000). In this thesis, a new approach based on a genetic algorithm is implemented in such a way that the new approach fulfils all of the above requirements. The

algorithm is tested on wide range of different data sets including Tic Tac Toe, Zoo, and 10000×8. These data sets come from the University of California at Irvine (UCI) machine learning repository (http://archive.ics.uci.edu/ml/datasets.html).

3.3 Requirements for Developing Association Rule Mining Algorithm

There are four main requirements for developing an efficient association rule mining algorithm. A set of techniques is needed which fulfils the following requirements:

1) Users do not need to specify an appropriate threshold value for mining rules although they have no information regarding the data set.
2) It will not search all the frequent item sets to generate rules from those frequent item sets.
3) It will not generate trivial and misleading rules (Martin et al. 2014).
4) It will extract a set of rules which are easy to understand and interesting.

Most of the existing classical algorithms for mining association rules are based on a support-confidence framework. This framework consists of two sub processes: finding all frequent item sets and generating rules from those frequent item sets based on a user defined support value and a confidence value, respectively. Several authors have noted that these algorithms do not satisfy the above requirements (Yan et al. 2005; Yan et al. 2009; Martin et al. 2014).

Recent GA based approach named ARMGA (Yan et al. 2005; Yan et al. 2009), uses conditional probability as a fitness function to extract high quality Boolean association rules (BARs). This algorithm uses only one evaluation criterion to measure the quality of the rules. Recently, some researchers have framed the association rule mining problem as a multi-objective problem in order to extract a set of rules which is easy to understand and interesting (Martin et al. 2014; Alatas & Akin 2008a; Ghosh & Nath 2004).

Based on the design factors and data sets, new multi-objective evolutionary algorithms are implemented, which jointly optimize multiple objectives to mine a reduced set of BARs without considering user defined support and confidence values. The generated rules are interesting, easy to understand and having a good trade-off among the number of rules, support, confidence and other objectives of the data sets. To accomplish this, the recent ARMGA is extended by the new approaches in order to perform an evolu-

tionary learning and condition selection and maximizes different objectives: lift, net confidence, conditional probability, interestingness and so on. For evaluating the performance of the proposed methods, the experimental studies are carried out on different real world data sets, and compare the performance of the proposed approaches with different GA based approaches (Yan et al. 2005; Yan et al. 2009; Qodmanan et al. 2011) and classical algorithms such as Apriori (Agrawal & Srikant 1994; Borgelt 2003), Eclat (Hipp et al. 2000; Zaki 2000) for mining BARs. The scalability of the proposed methods is studied and finally the rules which are generated by the proposed methods are analysed.

3.4 New Evolutionary Algorithms for Mining Frequent Patterns

3.4.1 GeneticMax: A New Evolutionary Algorithm for Improving Level by Level Searching Method Named Apriori

Genetic algorithm (GA) is an adaptive heuristic search method and it is applied in optimization problems. It is used as a general search approach with robustness and high scalability (Du et al. 2009; Yan et al. 2009). Due to its high scalability, a new approach named GeneticMax, which is based on GA, is designed in such a way that it decreases time complexity for mining frequent patterns from a large data set.

To generate maximal frequent item sets (MFI) from a large data set is the most time consuming task in the present day. In this research, an evolutionary approach is presented for finding maximal frequent item sets from large data sets by using the principles of Genetic Algorithm (GA). The search strategy of the new approach uses a lexicographic tree that avoids level by level searching, which finally reduces the time required to mine maximal frequent item sets in a linear way.

This algorithm also includes bitmap representation of nodes in a lexicographic tree and from the superset-subset relationship of nodes, it identifies frequent item sets.

The significant difference between Apriori and GeneticMax is that it randomly generates the chromosome and if the generated chromosome is in the positive boundary area, then it prunes all the subsets of that chromosome. Through this technique, it reduces the cost of calculation of support value of all the subsets of generated chromosomes. Whereas the Apriori algorithm calculates the support value of all the chromosomes in each level and prune those chromosomes which do not satisfy a user define support val-

ue at that level. From the above discussion, it can be concluded that the new approach is more efficient than Apriori algorithm.

The length of a frequent item set depends on its relationship among the item sets. The major advantage of this approach is that it performs a global search and its time complexity is less than that of other algorithms. Another advantage is that it generates frequent item sets independently of the size of the data sets.

This work differs from existing research (Kabir et al. 2014) in the following aspects: 1) Unlike Apriori, this approach uses a lexicographic tree (Agarwal et al. 2001) as a search space and it does not need to enumerate frequent item sets level by level; 2) For comparative analysis, Apriori and this approach are applied on different real data sets as well as synthetic data sets. Finally, the results are compared with the results of Apriori algorithm. 3) Unlike a Boolean based approach (Salleb et al. 2002) and FP- growth algorithm (Han et al. 2000), this approach does not need memory for loading a lexicographic tree which avoids the large consumption of memory space. This technique dramatically reduces the time for accessing a large data set to calculate the support value of unnecessary individuals to find frequent item sets. Although it is invented a long time ago but still Apriori is one of the famous algorithms and it performs better than other existing algorithms like Eclat, Partition, and DIC, especially when the support value is set high (Hipp et al. 2000). The performance analysis of Apriori and other existing famous algorithms of the present day is shown by Hipp, Guntzer and Nakhaeizadeh (Hipp et al. 2000). For this reason, an Apriori algorithm is chosen for comparison with the newly designed approach. CPU time (Run time) is needed by the existing mining approaches for calculating support values of examined nodes. The efficiency of an algorithm depends on how many numbers of frequent or infrequent item sets it considers to get the final solution i.e. maximal frequent item sets. In this research, thorough experiments demonstrate how many numbers of nodes i.e. item sets are considered by GA based approach and the results are compared with Apriori algorithm for different support values and data sets.

3.4.2 Hybrid GeneticMax: Improving GeneticMax Algorithm by Introducing a New Algorithm Named Hybrid GeneticMax

The early developed method, GeneticMax, is improved and extended by another approach named Hybrid GeneticMax. Three main features are embedded by the new approach:

1) it sorts out infrequent items from 1- item sets,
2) there is a superset-subset relationship in both positive and negative boundaries in a lexicographic tree for pruning invalid chromosomes, and
3) the use of a genetic algorithm which uses a global search mechanism. The purpose of sorting out infrequent items from 1-item sets is that, if an item is infrequent then all of its super item sets are also infrequent. Through this technique, the search space is dramatically reduced by this approach for finding the solution. The aim of this new approach is converging to a solution as fast as possible, especially if 1-item sets contain a reasonable amount of infrequent items and the solution resides in the deep level of the lexicographic tree instead of near the root. A full experiment of the new approach on different data sets are conducted which demonstrates the ability of this approach to yield solutions rapidly by accessing the data sets for a few number of nodes in a lexicographic tree.

From the previous discussion, it can be concluded that, all the nodes in each level of a lexicographic tree are tested by Apriori algorithm and those nodes of a level which do not satisfy a user defined support value are pruned. In GeneticMax, if it generates an individual X in any level which satisfies a user defined support value, then all other subsets of X in any level are automatically pruned. This mechanism is also true the other way around: if it generates an individual Y on any level which is infrequent i.e. which does not satisfy a user defined support value, then all the supersets of Y in any level of a lexicographic tree are automatically pruned. The Hybrid GeneticMax embeds all the features of the GeneticMax algorithm including local search mechanism for finding infrequent item sets from 1- item sets of a large data set.

3.5 Mining Association Rules for Both Frequent and Infrequent Items Using PSO

PSO is another optimization technique based on the intelligence and movement of swarms/ particles. To solve a problem PSO normally applies social interaction. In mul-

tidimensional search space, each individual within the swarm is represented by a vector. To determine the next movement of the particle each vector is assigned by a vector called velocity vector. The velocity is updated by each particle depends on the current velocity and the best position it has explored. This process is iterated by fixed number of times or it will continue until a minimum error is achieved. This simple model efficiently works for difficult optimization problems (Alatas & Akin 2008b; Khan et al. 2010; Merwe & Engelbrecht 2003; Eberhart & Shi 2001).

Traditional Association rule mining approaches include two steps:

1) mine frequent item sets based on a user defined support value from large data sets, and
2) generate association rules or correlation relationship among a large set of data items.

The traditional approaches reveal valid association rules by using support and confidence values of item sets in a database. To prune the search space these approaches use a minimum support value as a threshold. Two main problems arise because of using such mechanisms:

1) If users set minimum support value too low then it increases the computational complexity such as generation of candidate item sets, the complexity of designing a large number of tree nodes, testing of nodes and so on. Finally, it generates a large number of association rules and traditional algorithms suffer poor performance because of these large number of rules.
2) If users set minimum support value too high, many interesting rules with low support values are missed. Such association rules with low support values are important to discover the relationship among expensive items such as diamond or gold necklaces, ear rings, bracelets. These rules are also important for identifying such web documents which are identical or similar.

Recently, some researchers develop algorithms to mine association rules without a minimum support value constraint (Cohen et al. 2001; Wang et al. 2001; Xiong & Tan 2003). These approaches use confidence based pruning mechanisms instead of support based pruning techniques. Support free association rule mining techniques discover high, cross and low support based rules. Item sets with high support values are well known

patterns. Patterns containing items with cross support value have the poor correlation. On the other side, patterns with a low support value provide precious insights.

In data mining research, generating frequent items from large data sets is one of the important issues and the key factor for implementing association rule mining tasks. Mining infrequent items such as relationships among rare but expensive products is another demanding issue which has been shown in some recent studies. Therefore, this study considers user assigned threshold values as a constraint which helps users to mine those rules which are more interesting for them. In addition, in real world users may prefer to know relationships among frequent items along with infrequent ones.

The particle swarm optimization algorithm is an important heuristic technique in recent years and this study uses this technique to mine association rules effectively. If this technique considers user defined threshold values, interesting association rules can be generated more efficiently. Therefore, this study proposes a novel approach which uses a particle swarm optimization algorithm to mine association rules. The implementation of the search strategy includes bitmap representation of nodes in a lexicographic tree and from the superset-subset relationship of the nodes it classifies frequent items along with infrequent item sets. In addition, this approach avoids extra calculation overhead for generating frequent pattern trees and handling large memory which stores the support values of candidate item sets.

The main contributions of this work are as follows:

1) A new algorithm is proposed including the traditional particle swarm optimization algorithm to mine association rules from frequent and infrequent item sets,
2) These item sets are searched from a lexicographic tree which are based on user defined threshold fitness values, and
3) This scheme mines interesting rules not only for two or three item sets but also for large item sets.

3.6 New Multi-Objective Evolutionary Algorithms for Extracting Reduced Sets of Boolean Association Rules

Based on the design factors and data sets, in this section, two new multi-objective evolutionary models, named ARMGAAM and MBAREA, which are designed for mining a reduced set of BARs are described. The former method uses a re-initialization technique

along with an adaptive mutation method whereas the latter uses a class based mutation method along with a best population technique. Both methods discover a reduced set of BARs from different data sets with a good trade-off among the number of generated rules and different measures.

3.6.1 ARMGAAM: Multi-Objective Evolutionary Algorithm Using Adaptive Mutation Method

Association rule mining is the process of discovering useful and interesting rules from large data sets. Classical association rule mining algorithms depend on a user specified minimum support and confidence values. These constraints introduce two major challenges in real world applications: exponential search space and a data set dependent minimum support value. Data analysers must specify a suitable data set dependent minimum support value for mining tasks although they might have no knowledge regarding the data set and these algorithms generate a huge number of unnecessary rules. To overcome these kinds of problems, recently several researchers framed association rule mining problem as a multi-objective problem (Martin et al. 2014; Mukhopadhyay et al. 2014).

The final population of the existing ARMGA (Yan et al. 2009; Yan et al. 2005) model miss some useful rules which are better and generated in some intermediate generations because of using the standard genetic operators. Moreover, these approaches generate too many unnecessary rules because of using weak constraint such as relative confidence as a fitness function.

To evade these issues, in this research work a new multi-objective evolutionary algorithm for mining BARs named ARMGAAM is proposed, which generates a reduced set of association rules and optimizes several measures that are present in different degrees based on the data sets are used. To accomplish this, the proposed method extends the existing ARMGA model for performing an evolutionary learning, while introducing a re-initialization process along with an adaptive mutation method. Moreover, this approach maximizes conditional probability, lift, net confidence and performance in order to obtain a set of rules which are interesting, useful and easy to comprehend. The effectiveness of the proposed method is validated on different real world data sets.

3.6.2 MBAREA: Improving Traditional GA Based Approach for Mining Boolean Association Rules

This section describes the proposed method for obtaining a reduced set of interesting association rules with a good trade-off between the coverage and the number of generated rules, considering three objectives conditional probability, lift and interestingness. This proposal extends the existing ARMGA and ARMMGA algorithms for performing an evolutionary learning and introduces two new components: class based variable adaptation operator and best population.

In order to store all the non-dominated rules which are generated in the intermediate generation of a population, provoking the diversity of the population, and increasing the coverage of data sets, a new class based mutation approach along with best population method are designed. The mutation operator is used to keep the diversity from one generation of a population to the next one. The mutation changes one or more genes of a chromosome with respect to a mutation probability, mp.

Existing GA based approaches such as ARMGA and ARMMGA, follow fixed mutation probability and randomly mutated the chromosomes. Although, low mutation probability is used by these methods, few high quality chromosomes are mutated due to the random function. For this reason, some top quality chromosomes get less chance for the future generation of a population.

To prevent this problem and to give more chance to the best chromosomes for the future generation of a population, the whole population are classified into δ, based on a fitness value of each chromosome. Top class chromosomes have a higher fitness value but assign with a low mutation ratio whereas low class chromosomes are mutated with high mutation probability. Through this approach high class chromosomes take part for future generation of a population. Best population (BP) keeps all the non-dominated rules which are generated in the intermediate generation of a population. Moreover, BP will be updated with the generation of a new population following the non-dominance criteria. This process helps to increase the coverage of a data set and performs enhanced exploration of the search space.

3.7 MSGA: A New Evolutionary Algorithm Based on Multiple Seeds

In order to address the major challenges and issues raised by single seed based genetic algorithm, a novel framework named MSGA (Multiple Seeds Based Genetic Algorithm) is presented for obtaining strong search ability. The novel features of this method are as follows:

1) *m-Domain* Model. This proposed method subdivides the whole solution space into an *m-number* of same size domains. The purpose of dividing the whole solution space is to get seeds from each domain and to maintain diversity for generating an initial population.
2) *m-Seeds* Selection Process. This proposed approach uses the *m-seeds* selection process where *n-number* of chromosomes are generated from each domain. From the members of a domain, this process selects only one high fitness value chromosome as a seed. This chromosome is used as a seed for that domain.
3) Initialize Population based on *m-Seeds*. Based on a seed chromosome, the next step is to generate *n-number* of individuals by randomly changing any position of a seed chromosome. Through this technique, m seeds generate a $m \times n$ number of individuals. These individuals are used as an initial population for MSGA.
4) Method Implementation and Evaluation. In order to demonstrate the effectiveness of the proposed approach, a large number of studies is carried out on association rule mining approaches and different crossover and mutation operators. To demonstrate the feasibility of the proposed method, a number of experiments are conducted to mine a reduced set of interesting association rules by optimizing conditional probability using different crossover and mutation operators. To compare a single seed based approach with the proposed method, the same set of experiments is applied on different data sets for mining BARs using different crossover and mutation operators by initialising a population using a single seed. Experimental results in Chapter 5 show that a multiple seeds based method demonstrates satisfactory performance over different single seed based methods.

The major focus of this algorithm is to apply the multiple seeds based generation mechanism to generate diversified initial population with good coverage into the evolutionary process, which generates a large amount of high quality rules. This process also helps to

automate selection of a seed chromosome without depending on a data set. This approach is applied to ensure that an evolutionary algorithm is not trapped into a local optimum in an early stage and ensures multiple convergences on a whole solution space. Thus, an overall global optimum is achieved. In the following chapters, all the characteristics of a multiple seeds based genetic algorithm are described (see Chapter 4).

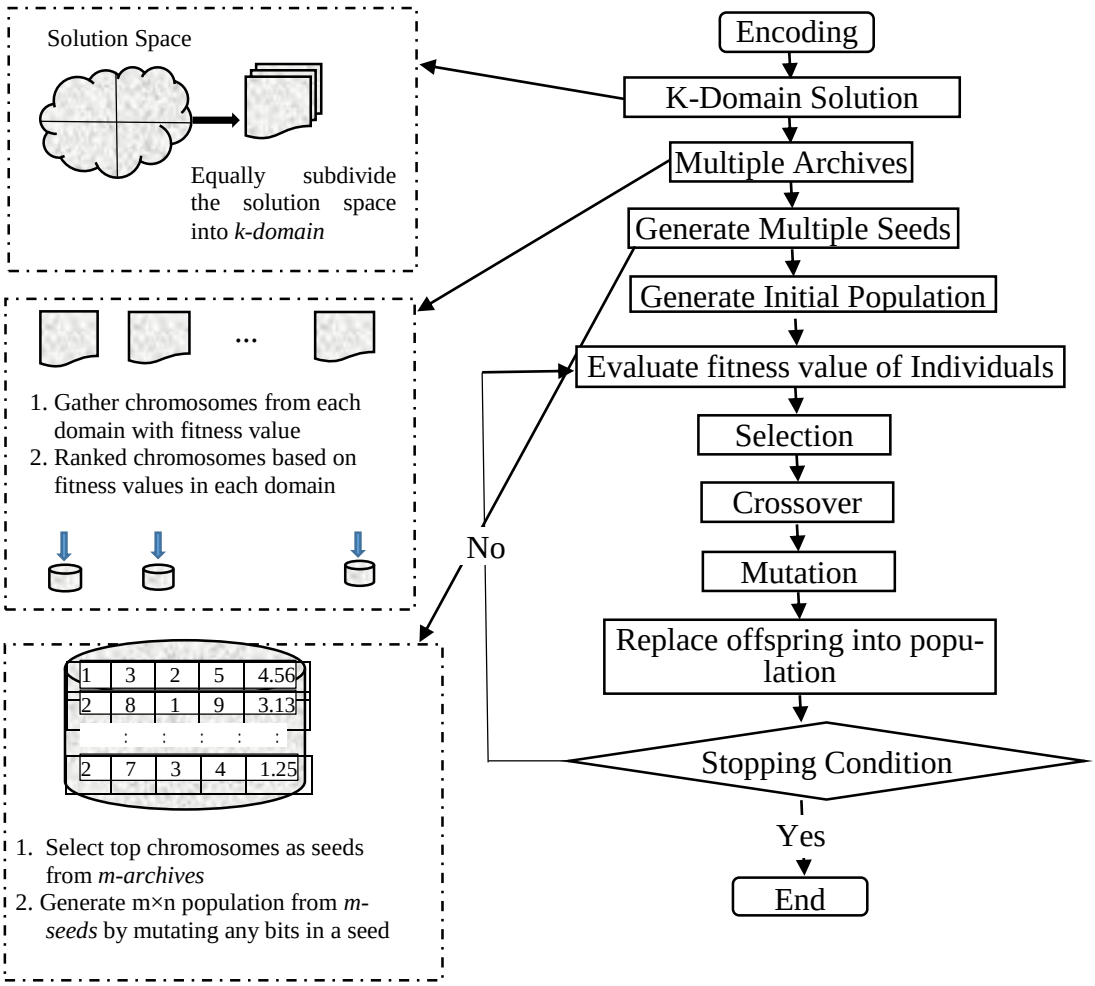


Figure 7: The architecture of MSGA

The traditional genetic algorithm uses a single seed to generate an initial population. The basic idea of a single seed based genetic algorithm (SSGA) is to randomly select a chromosome from a large solution space for generating an initial population. Because of random selection, SSGA could face premature convergence problem and extract a small number of high quality rules from a large data set.

On the other hand, some seeds may have a low fitness value but could generate a large number of high quality rules since it explores a huge area of a large solution space. The random selection of a seed chromosome and generating an initial population based on

that chromosome cannot guarantee whether that population cover the whole solution space or not. As an initial population has significant effects on obtaining best results after several generations, so the population diversity including good coverage of a large solution space is important for the generation of an initial population for balancing the exploration and exploitation search. Thus, a global optimum can be achieved.

The basic idea of this research is to equally divide the whole solution space into *m-domain*. From each domain, this method generates *n-number* of individuals. The individuals which are randomly generated from a domain are stored in an archive. Therefore, *m-domains* generate *m-archives*. Chromosomes of each archive are ranked based on the fitness values of those chromosomes. A chromosome of a high fitness value has a higher rank than the chromosome of a low fitness value. From each archive, top ranked chromosome is selected as a seed. By mutating any bits of a seed, each seed generates *n-number* of individuals. Therefore, *m* seeds generate $m \times n$ individuals which are used as an initial population for multiple seeds based genetic algorithm. The architecture of MSGA is shown in Figure 7.

3.8 Chapter Summary

This chapter has presented comprehensive justification of the proposed methods. The requirements for developing the frequent pattern and association rule mining tasks had explained, which followed the explanation of the main features and the theoretical aspects of the proposed methods for mining maximal frequent item set and Boolean association rules. In order to address the major challenges and issues raised by a single seed based genetic algorithm, the novel features of Multiple Seeds Based Genetic Algorithm (MSGA) have been explained. The architecture of MSGA was also presented in this chapter.

The next chapter will describe the framework and the underlying concepts of the proposed algorithms.

Chapter 4 - Implementation of Methodologies

4.1 Introduction

Following on from the research methodologies chapter which provided the overview, the specific details of methodologies are described in this chapter.

Each approach describes the underlying concepts and structure which are used for extracting frequent patterns and association rules from large data sets. In addition, each algorithm is described by the pseudo code. Initially, the problem of mining maximal frequent item sets is addressed and the pseudo code of the GeneticMax algorithm is explained through section 4.2. The basic notions and the structure of the Hybrid GeneticMax algorithm are described in section 4.3. The underlying concept and the framework of the PSO based method for mining association rules for both frequent and infrequent items are described in section 4.4. The basic concepts, objectives and the flowchart of the proposed algorithms for mining Boolean association rules, named ARMGAAM and MBAREA, are explained in sections 4.5 and 4.6, respectively. Finally, the technique for encoding, generating an initial population from multiple seeds along with the pseudo code of multiple seeds based genetic algorithm is described in section 4.7.

4.2 Mining Frequent Patterns Using GeneticMax

4.2.1 Problem Definition

A huge number of frequent patterns are generated from big data sets which satisfy a user defined threshold value especially when users assign a lower value for min_supp. Generating an enormous number of frequent item sets from large data sets is a major challenge in a frequent pattern mining task. If an item set is frequent, all of its sub item sets are frequent. To solve this problem, researchers proposed closed and maximal frequent pattern mining task (Borgelt 2012). For this study, maximal frequent pattern mining task is considered.

Definition (Maximal Frequent Pattern Mining).

An item set ε is a maximal frequent item set in a data set D, if ε is frequent and there exists no superset η which is frequent in a data set D such that $\varepsilon \subseteq \eta$. A frequent item set is called maximal, if all of its sub item sets are frequent whereas all of its super item sets are infrequent.

In this research, maximal frequent item sets (MFI) are mined from a large data set *D*, where a user defined support value acts as a constraint.

4.2.2 Lexicographic Tree

The research problem here is to find maximal frequent item sets from large data sets using Genetic Algorithm. Item set *I* consists of *n* items, i.e. $I = \{i_1, i_2, i_3, ..., i_n\}$. X_k represents an item set containing *k-items*, where $k = 1, 2, \ldots, n$ and $X_k \subseteq I$. If $k=1$, then X_k contains a *1-item*, i.e. $X_1 = \{i_1\}$. If $k=2$, then X_k contains *2-items*, i.e. $X_2 = \{i_3, i_4\}$, and so on.

In this research, we will consider a search space which consists of all feasible solutions. A Lexicographic tree (D. Burdick et al. 2005; Huang et al. 2004) is the search space for GeneticMax. A Lexicographic tree maintains the lexicographic ordering of items *I* in a data sets *D*. If an item *i* occurs before item *j* in a data set *D*, then it maintains the lexicographic ordering, i.e. $i \leq_L j$. If two subsets S_1 and S_2, where $S_1 \subseteq S_2$ and $S_1, S_2 \in S$ then it maintains the following lexicographic order: $S_1 \leq_L S_2$. There is no lexicographic ordering relationship between two subsets S_1 and S_2 if S_1 and S_2 are disjoint subsets.

Figure 8 shows an example of a lexicographic tree which considers lexicographic ordering for four items. The root of the tree is an empty set and each *k-level* contains *k-items*. In each level, *k-item* sets maintain lexicographic ordering with the tail nodes containing items lexicographically larger than elements of the head node. The support value of the head node is more than that of the tail node. It can be seen that the nodes closer to the root are more frequent than those far from the root. There is a non-linear line (called a cut) in the tree which separates frequent item sets from infrequent ones. The nodes which are above the cut are frequent item sets and the elements below this cut are infrequent ones.

For GeneticMax, a new tree is introduced which is based on a user defined support value. The line is defined by a user defined support value and the area above the line is referred to as a positive area and the area below the line is referred to as a negative area. All the nodes in a positive area are frequent whereas all the nodes in a negative area are infrequent.

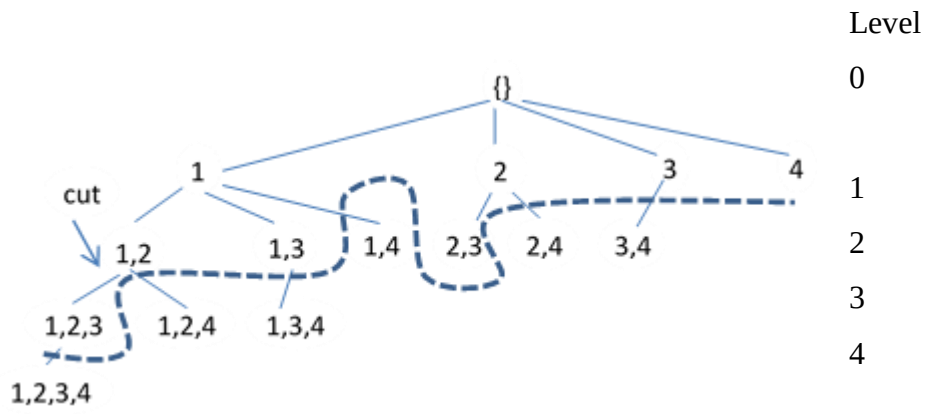


Figure 8: Lexicographic tree of four items

If the tree of Figure 8 is redesigned, the lexicographic tree of these four items would be as follows:

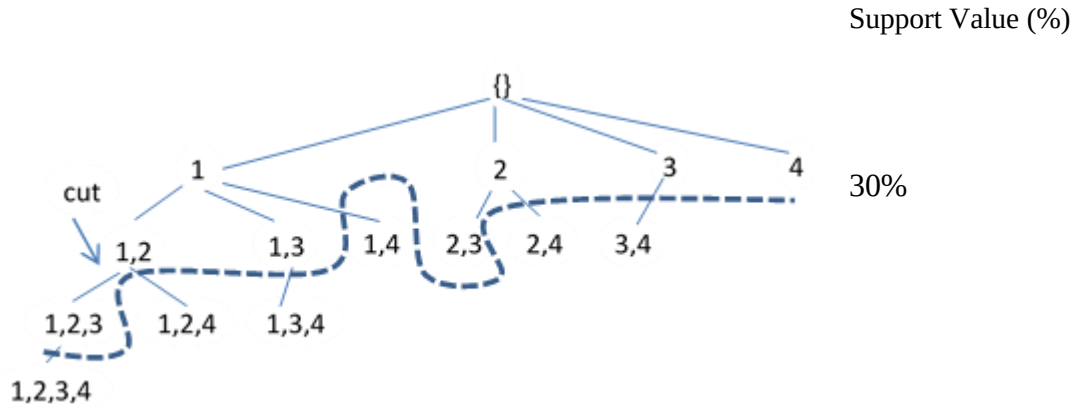


Figure 9: Lexicographic tree of four items based on a user defined support value

In Figure 9, the nodes within the positive boundary area have a minimum support value which is 30%. GeneticMax introduces an array which stores the frequent item sets (called FIs), and among the frequent item sets, the set containing the largest number of items is called the maximal frequent item set. This maximal frequent item set (stored in another special array) is called MFI. This algorithm searches frequent nodes within a positive area and tries to converge to a solution: finding maximal frequent item sets as early as possible. Figure 9 verifies Lemma 1 (see section 2.2.1.1), where there are 4 items, and it enumerates 2^4-1=15 nodes including the root node. With Apriori algorithm, one would test all the nodes in a specific level and generate candidate item sets. The generation of candidate item sets needs a long time for finding maximal frequent item sets. For example, in Figure 9 it tests the item sets {1},{2},{3},{4} in level 0 and finds that all the item sets are frequent since these nodes satisfy the minimum support value. Then it

goes to the next level to scan the data sets to get the support values of {1,2},{1,3}, {1,4}, {2,3},{2,4},{3,4} and so on. On the next level it prunes the item sets {1,4},{2,4},{3,4} since these nodes have support values which are less than the user defined support value. Unlike Apriori algorithm, GeneticMax does not need to test all the nodes, saving a significant amount of time even when a data set is very large. For example, if the initially generated item set is {1, 2, 3} then it scans the data set and calculates the support value. If the support value of the generated item set {1, 2, 3} is ≥ 30%, then it stores this item set in a frequent item set array called FI_Superset_Member. Future scans will not look the data sets for {1},{2},{3},{1,2}, and {1,3} since these item sets are the subsets of the previously generated item set {1,2,3}. If the generated item sets are {1},{2},{3} or {1, 2} then it always checks the array FI_Superset_Member. If it finds any superset in FI_Superset_Member, then GeneticMax will discard these subsets, substantially reducing the time for scanning the data sets to calculate the support values.

Lemma 3: If *Y* is a superset of an item set *X*, i.e., $X \subseteq Y$ and if *Y* is a frequent item set, then it can be claimed that *X* is a frequent item set.

For example: {1,2,3} is a superset of item set {1},{2},{3},{1,2} and {1,3}. As GeneticMax uses the principles of Genetic Algorithm and follows the global search mechanism, a superset could be generated before generating a subset. In this example, if {1, 2, 3} is generated before its subsets (and stored in the array FI_Superset_Member), then all other generated subsets will be discarded.

Lemma 4: If *Y* is a superset of an item set *X*, i.e., $X \subseteq Y$ and if *X* is an infrequent item set, then it can be claimed that *Y* is an infrequent item set.

For example, if the initially generated chromosome is {1, 4} and the support value of this item set is <30%, then it is stored in a non-frequent item set array called NFI. If the next generated item set is {1, 3, 4}, the algorithm will check the NFI array, and if it finds any subset in this array, GeneticMax will discard the item set {1, 3, 4} for any future calculations.

Lemma 5: If *Z* is a superset of an item set *X, Y*, i.e., $X,Y \subseteq Z$ and if *Z* is an infrequent item set, then it cannot be concluded whether *X* or *Y* is an infrequent item set.

Lemma 5 is slightly different from Lemma 3. With the previous example, if {1,2,3} is a frequent item set then all of its subsets must be frequent, i.e., {1},{2},{3},{1,2},{1,3},

{2,3} are all frequent item sets. But if {1, 2, 3} is an infrequent item set then it cannot be conclude that all of its subset are infrequent. In Figure 9, {1, 4} is an infrequent item set but its subsets {1} and {4} are frequent item sets.

Lemma 6: If *Z* is a subset of item sets *X* and *Y*, i.e. $Z \subseteq X$, $Z \subseteq Y$ and if *Z* is a frequent item set, then it's supersets *X* and *Y* could be either frequent or infrequent item sets.

For example, in Figure 9, item set {1} is frequent although its superset {1, 3} is frequent and its superset {1, 4} is infrequent.

The main idea of GeneticMax is to find maximal frequent item sets, while converging to a solution as fast as possible. It subdivides a whole lexicographic tree into two sub-areas based on a user defined support value. GeneticMax can generate any chromosome in any sub region. If it finds any superset in a positive boundary area, then it follows Lemma 3 and prunes all of its subsets. But if it finds any subset in a negative boundary area, then it follows Lemma 4 and prunes all of its supersets.

The main advantage of GeneticMax is its ability to quickly converge to a solution, and find all the supersets in a positive boundary area closer to the cut as fast as possible. In the above example, if {1,2,3} is generated before all of its subsets ({1},{2},{3},{1,2}, {1,3}, {2,3}) and found to be a frequent item set, then it will discard those subsets (which are also frequent item sets). If the next generated item set is {1, 3, 4} it will check the NFI_Subset_Member array and does not find any subset there. GeneticMax will scan the data sets for this item set to find its support value and store it in NFI discarding all the supersets of {1, 3, 4} in future scans.

4.2.3 Description of GeneticMax

4.2.3.1 Mapping Item Sets to Chromosomes

GeneticMax maps item sets onto a chromosome code. Each node in the lexicographic tree represents different item sets and all the nodes in the tree get a unique chromosome code. The main features of chromosome coding are,

1) It calculates the support value easily since GeneticMax uses bitmap representation of the data sets.

2) It generate all possible nodes.

If there are n items, it enumerates (2^n-1) item sets or nodes in the lexicographic tree. GeneticMax can generate (2^n-1) nodes if required.

The length of a chromosome is fixed. If a data set contains n items, the length of all generated chromosomes is always n. The chromosomes look like the following:

V_{item_1}	V_{item_2}	⋯	V_{item_n}

Figure 10: Mapping items onto chromosomes $V_{item_{1\ldots n}} \in [0, 1]$

Since the original database contains a large number of items, a transaction t_1 of a data set D is of the form, $t_1 = \{i_{11}, i_{12}, \ldots\ldots, i_{1k}, \ldots i_{1n}\}$. The value of item $i_{1j}, j \in 1 \ldots n$ is either 1 or 0, depending on whether it is present or absent in a transaction t_1.

4.2.3.2 Population Generation

The population of a genetic based system is generated as follows: For the first individual, the whole domain is considered in a lexicographic tree. For the following individuals, those item sets are considered which are not classified in frequent or infrequent ones. When the individual is generated, it is classified as frequent or infrequent ones through the use of a user defined support value. Through this technique the search space become narrower for the generation of the next population.

4.2.3.3 Genetic Operators

To improve the quality of the next individual, crossover and mutation operators are used to transform one individual into another one. At the initial stage of population generation, two parent individuals are selected randomly from the domain of item sets in a lexicographic tree and after applying crossover operator two new offspring are generated. Mutation changes a single bit randomly in each segment of the individuals for the improvement of new offspring.

4.2.3.4 Procedure of GeneticMax

Step 1: Set a generation number.

Step 2: Generate population of GeneticMax.

Step 3: Check the FI_Superset_Member and NFI_Subset_Member array for supersets and subsets of this generated chromosome.

Step 4: If it finds any supersets or subsets in FI_Superset_Member or

NFI_Subset_Member, respectively, then go to Step 2.

Step 5: Compute a fitness value of individuals according to their support values in a data set *D*.

Step 6: Perform FI_Member_Add, and if any frequent item sets are found then update FI_Superset_ Member.

Step 7: Perform NFI_Member_Add, and if any infrequent item sets are found then update NFI_Subset_Member.

Step 8: Go to Step 3 with newly generated chromosomes until it exceeds the generation number which was set by Step 1.

4.2.3.5 Mining the Superset in a Positive Boundary Area

For an item set *X*, if there is any subset of *X* in FI_Superset_Member, then this method is called to replace that subset by its superset *X*. This method is also applicable if *X* is a new frequent item with no subset in FI_Superset_Member.

```
//Invocation: FI_Member_Add(I_F , FI_Superset_Member)
    1. If any subset of  I_F  is in FI_Superset_Member
    2.       Delete the Subset of I_F
    3.       Add I_F in FI_Superset_Member
    4. Else add I_F in FI_Superset_Member
```

4.2.3.6 Mining the Subset in a Negative Boundary Area

For an item set *X*, if there is any superset of *X* in NFI_Subset_Member, then this method is called to replace that superset by its subset *X*. This method is also applicable if *X* is a new infrequent item and it has no superset in NFI_Subset_Member.

```
//Invocation:NFI_Member_Add(I_IF, NFI_Subset_Member)
    1. If any superset of is in NFI_Subset_Member
    2.       Delete the Superset of I_IF
    3.       Add I_IF in NFI_Subset_Member
    4. Else add I_IF in NFI_Subset_Member
```

4.2.3.7 Pruning Methods of GeneticMax

Check_Member_for_Item function incorporates three techniques:

1) Superset Checking Techniques

Checking to see whether a given chromosome is a superset in a positive boundary area. Further pruning happens if a given item set is not a superset in the positive boundary area.

2) Subset Checking Techniques

Checking to see whether a given chromosome is a subset in a negative boundary area. Further pruning happens if a given item set is not a subset in the negative boundary area.

3) Unchecked item set checking techniques

If an item set is neither a superset in a positive boundary area nor a subset in a negative boundary area, then this item set is referred to as an "unchecked" item set and needs to be tested. For this unchecked item set, GeneticMax scans the data sets and sets the item set in FI_Superset_Member or NFI_Subset_Member according to a user defined support value.

```
//Invocation:Check_Member_for_Item(I,
FI_Superset_Member, NFI_Subset_Member)

1. If any superset of I is in FI_Superset_Member
2.       Discard I
3. Else if any subset of I is in NFI_Subset_Member
4.           Discard I
5.       Else scan the database to calculate support value
             for I
6.       If support value ≥ user-defined support value
7.           Invoke FI_Member_Add
8.       Else Invoke NFI_Member_Add
```

4.2.3.8 Fitness Function

The fitness function of this proposed method provides fitness value of an individual which is equal to the support value of an item set i.e. $f_{individual_i} = support$, where $f_{individual_i}$ is the fitness value of individual i. When an individual is generated, the support value for that individual is counted from the data set. If the fitness value of an item set satisfies a user defined support value i.e. $f_{individual_i} \geq \min\_supp$, then this item set is classified as frequent and stored in an array called FI_Superset_Member. Otherwise, it will save in an array, called NFI_Subset_Member. Members of FI_Superset_Member array are the frequent item sets of a data set and are always superseded by the supersets of member item sets. Similarly, members of NFI_Subset_Member array are the infrequent item sets of a data set and are always replaced by the subsets of member item sets.

A prototypical genetic algorithm based scheme is followed by the proposed method. Minimum support value (min_supp), number of generations (NbGen), mutation rate (MR), crossover rate (CR), a data set (TuplesNb) are the inputs of the algorithm.

Fitness (item set) **Function**

```
Temp_Fitness = SupportCount(item set)
// Count the support value of item set  from given data set
if Temp_Fitness ≥ min_supp then
   return +Temp_Fitness, item set
else
   return –Temp_Fitness, item set
```

MFItemsets **Function**

Input: min_supp, NbGen, MR, CR, data set composed of TuplesNb

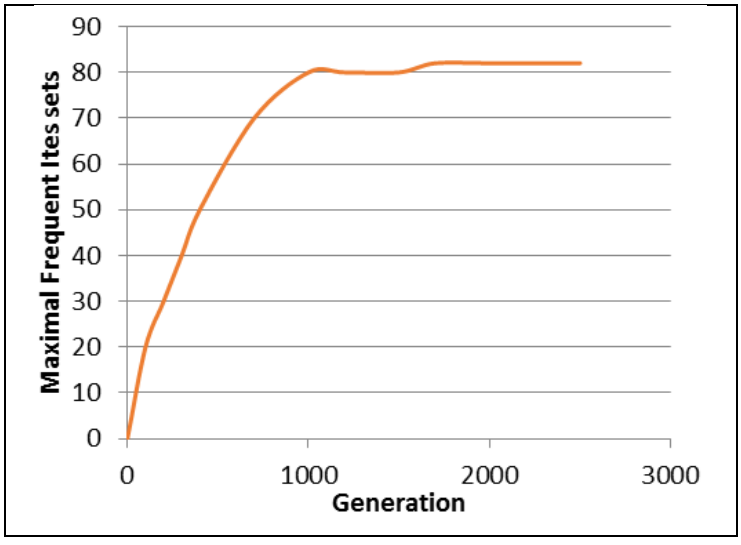
Output: Maximal frequent item sets MFI, NbTestingNodes

```
Generate a random population
While i ≤ NbGen do
  Select two parents from generated individuals and applying crossover
  and mutation operators to get new two offspring
  foreach individual
    check FI_Member_Add array, if this individual or any of its subset
          is in this array
    check NFI_Member_Add array, if this individual or any of its
         superset is in this array
    If none of the above arrays contain this individual or any of its
      subsets or supersets then
```

```
    Fitness_Value = Fitness (individual)
    NbTestingNodes++
    if Fitness_value > 0 then
      Update FI_Member_Add array
    else
      Update NFI_Member_Add array
  i++
  //FI_Member_Add array contains the latest maximal frequent item sets
  MFI = FI_Member_Add
return MFI, NbTestingNodes
```

4.2.3.9 Lifetime of GeneticMax

The lifetime of GeneticMax depends on user's selection of a generation. The higher the generation number the higher the probability for getting a correct solution. But there is a threshold value for a generation: after the threshold is reached the solution remains the same.



4.3 Improving GeneticMax Using Hybrid GeneticMax Approach

4.3.1 Basic Notions

Frequent item set mining is a famous data mining method originally developed for analysing market data. The main aim of this task is to find regularities in customers' shopping behavior in supermarkets, online shop and mailing orders of companies. Specifically, it tries to mine item sets that are frequently bought together by the customers from large transactional data sets. These sets of associated items help the organization to make decisions about which bundles of item sets should be offered, which bundles of items are popular to the customers and need to be arranged on the same shelf, or which bundles of products should be bought by the industry frequently, which will benefit industries by selling those products and so on. These days mining frequent item sets plays a vital role in different data mining tasks such as mining association rules, classification techniques, finding correlations among attributes of a data sets, clustering and many other interesting regularities among data.

Formal definition of a frequent item set mining is as follows: Given Item base $B = \{i_1,i_2,\ldots,i_{n-1},i_n\}$, which is a set of different items and a data set $D = \{d_1,d_2,\ldots,d_{m-1},d_m\}$, where *D* is a transactional, or other type of data set including car, zoo, and gaming data sets. For zoo data set, an item could be hair, feathers and so on. Top-left-square, bottom-right-square are the items for TicTacToe game. The item base represents the set of all items offered by the data set. Any subset of an item base, *B,* is referred to by the term *item set*. For example, if a transactional data set is considered, each transaction in a data set, *D,* is an item set, which is bought together by a customer on any day. Transaction id (*tid*) may be used to enhance each transaction. Item base *B* can be represented by the union of all transactions i.e. $\cup_{i(1\ldots m)t_i}$. The support value of an item set, *I,* is how many times this item set appeared in a data set. Let *x* be an item set. The support value of *x* is, $supp(x) = \left|\frac{x}{D}; x \subseteq d_i, d_i \in D\right|$.

An item set is frequent if its support value satisfies a user defined support value, min_supp i.e. $supp(x) \geq$min_supp.

The main problem with mining of frequent item sets is that often a large number of item sets is generated which satisfy min_supp threshold, especially for low min_supp value. To solve this problem researchers proposed different restrictions on the set of frequent

item sets. Mining maximal frequent item sets is one of the well known methods of those suggested proposals (Borgelt 2012). An item set x is maximal in a data set *D*, if *x* is frequent and there exists no superset *y* such that $y \supseteq x$, is frequent in a data set *D*.

As an illustration, Figure 11 shows a small transaction data set containing 8 transactions of item base $B = \{a, b, c, d, e\}$.

a)Transactions	
1:	{a,b,d}
2:	{c,d,e}
3:	{a,c,d}
4:	{a,b,c,e}
5:	{b,c,e}
6:	{b,c,d}
7:	{d,e}
8:	{a,b,d,e}

b) Frequent item sets (min_supp = 2)			
0 item	1 item	2 items	3 items
{}: 8	{a}: 4	{a,b}: 3	{a,b,d}: 2
	{b}: 5	{a,c}: 2	{a,b,e}: 2
	{c}: 5	{a,d}: 3	{b,c,e}: 2
	{d}: 6	{a,e}: 2	
	{e}: 5	{b,c}: 3	
		{b,d}: 3	
		{b,e}: 3	
		{c,d}: 3	
		{c,e}: 3	
		{d,e}: 3	

c) Maximal frequent item sets
{a,b,d}, {a,b,e}, {b,c,e}

Figure 11: a) A simple transaction database of 8 transactions containing 5 items, b) Frequent item sets based on a user defined threshold value, min_supp = 2 and c) maximal frequent item sets based on table b).

If the size of an item base *B* is α, then it will generate 2^{α} candidate item sets. It is computationally infeasible to determine the support value of all the candidate item sets and filter out the infrequent items, since a small supermarket or industry generally offers thousands of various items.

To make the search techniques efficient, different concepts have been proposed by researchers. One of the concepts which is still widely used is Apriori (Hipp et al. 2000) property. The main theme of this property is that, all the supersets of infrequent item sets are not frequent (Borgelt 2012). For this study, this property is used to find maximal frequent item sets. The difference is that Apriori considers level by level searching whereas Hybrid GeneticMax generates frequent item sets based on the property of a parent chromosome. The search space is that space which considers all feasible solutions. A Lexicographic tree (Bayardo 1998) is the search space for the Hybrid GeneticMax algorithm. Abstract representation of large item sets is done by this tree and it considers the lexicographic ordering, defined in the following way.

1) Each node of a lexicographic tree represents an item set.
2) If $I = \{i_1, i_2, \ldots, i_n\}$is an item set, where items $i_1, i_2, \ldots, i_n$ follows lexicographic ordering i.e. $i_1 \leq i_2$. Here $\{i_1, i_2, \ldots, i_{n-1}\}$ is the parent of item set *I*.

3) The root of the tree is an empty set.
4) The tree is the left most tree i.e. item sets are arranged from left to right.
5) A node which is closer to the root has a higher support value than a node which is further from the root.
6) There is a non-linear line in the tree called "cut" which separates infrequent item sets from frequent ones. This cut is defined in the tree based on a user defined threshold value i.e. min_supp.
7) Nodes above the cut are frequent item sets whereas nodes below the cut are infrequent item sets.

4.3.2 The Proposed Method

As mentioned above, the core of this study is an evolutionary algorithm where each individual represents an item set. The general view of the Hybrid GeneticMax algorithm, representation of each chromosome or individual, fitness function of each individual, generation of new individuals using genetic operators and item sets enumeration process are described in the following sections.

4.3.2.1 The Purpose of Using Genetic Algorithm

Genetic algorithm (GA), which simulates the natural behaviour of biological organisms, plays a vital role for this study. Genetic algorithm based techniques are robust and can be used to solve a wide range of problems including those which are hard to solve by other methods. Researchers concluded that, it is not guaranteed that GA always provide optimum solution to a problem rather it provides "acceptably good" solution to a problem which is solved by other method "quickly". Existing methods for solving a particular problem, can be improved by hybridizing with genetic algorithm (Beasley et al. 1993).

4.3.2.2 Hybrid GeneticMax Algorithm

The Hybrid GeneticMax algorithm is based on the theory of genetic algorithms. The structure of a lexicographic tree is based on a user defined threshold value (Kabir et al. 2014). This study will use this search space to find maximal frequent item sets. For this algorithm, data set, *D,* is the input and it returns maximal frequent item sets. In a brief, a data set, *D,* contains a large number of transactions i.e. $D = \{t_1, t_2, \ldots, t_{n-1}, t_n\}$ and each transaction contains items. The form of transaction t_1 is as follows: $t_1 = \{i_{11}, i_{12}, \ldots, i_{1j-1}, i_{1j}\}$. The presence or absence of an item i_{1k}, $k \in 1 \ldots j$ is represented by 1 or 0.

Algorithm Hybrid GeneticMax
Step 1: Find infrequent items from *1-item* sets and Initialize NFI array.
Step 2: Find maximal frequent item sets from *k-item sets* where *k*>1. 1. Set generation number NbGN = δ and nGN = 0 2. Generate initial population 3. **While** (nGN < NbGN) 4. Compute fitness value using fitness_function (individual) 5. **If** (fitness_function (individual) ≥ min_supp) 6. **If** subset of this individual is in FI array then replace it by the current individual 7. **Else** add individual in FI array 8. **Else If** superset of this individual is in NFI array then replace it by the current individual 9. **Else** add individual in NFI array 10. Select two parent individuals 11. Generate new individual, applying crossover and mutation operators on parent individuals 12. nGN++ 13. **end While** **End**

Figure 12: Hybrid GeneticMax algorithm

Firstly, it will use a local search to find infrequent items from *1-item sets* and initialize NFI array. NFI array will contain infrequent item sets. Secondly, it will use the genetic algorithm based approach to find maximal frequent item sets from *k-item sets* where *k*>1. The first step is to set a generation number by using the variable name (NbGN). Other parameters which should be used as an input along with given data sets *D*, are mutation rate (MR), crossover rate (CR) and minimum support value (min_supp).

Each individual is frequent or infrequent depending on the fitness value of that individual. If the given individual is frequent based on its fitness value, then the individual is stored in an array for further checking. The array of Hybrid GeneticMax algorithm is classified into two groups. 1) array of frequent item sets name FI and 2) array of infrequent item sets name NFI. Finally, the members of the FI array are the maximal fre-

quent item sets. The basic structure of Hybrid GeneticMax algorithm is shown in Figure 12.

4.3.2.3 Representation of Individuals

A transaction is an item set, which shows the presence or absence of items. An individual represents a transaction. A simple form of i^{th} individual is $individual_i$= attributes. The value of an attribute comes from an item set domain of a data set, *D*. Real codification is used to represent individuals. An individual of Hybrid GeneticMax algorithm is a *k-item set* i.e. *k-items* are present in the individual, where $k \geq 1$. If the size of an item base *B* is *n*, it will generate 2^n different item sets.

$Item_1$	$Item_2$	….	$Item_{n-1}$	$Item_n$

Figure 13: Representation of an individual for n- items

4.3.2.4 Generation of Population

At the initial stages of the Hybrid GeneticMax algorithm, an initial population is needed from which to generate the next population. The whole item set domain of a lexicographic tree is considered for generating initial individuals. Random generation of the initial population means the algorithm starts from any node of the lexicographic tree and classifies this item set as either frequent or infrequent based on a user defined threshold value. The initial population acts as parent individuals for generating next individuals. Genetic operators are applied on parent individuals to create new individuals.

4.3.2.5 Genetic Operators

Two essential operators named crossover and mutation are used to improve the quality of offspring. Parent individuals are selected randomly from the initial population. After random segmentation of parent individuals, a crossover operator is used to generate new individuals. A mutation operat is performed in the segmented region and new two offspring are generated. Figure 14 shows an example of the process of generating new offspring using genetic operators.

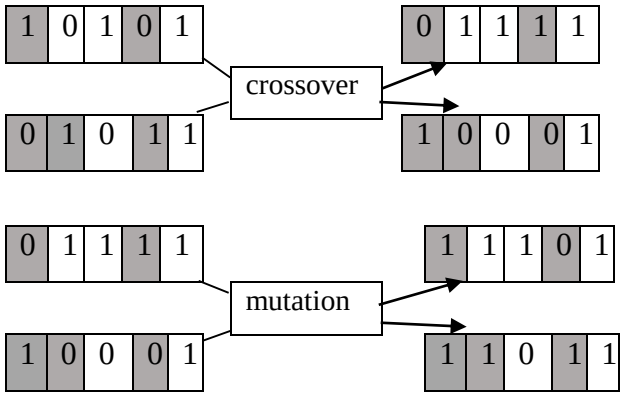


Figure 14: Process of generating new offspring using genetic operators

4.3.2.6 Fitness Function

The Lexicographic tree is classified into two areas based on a user defined threshold value, 1) frequent and 2) infrequent.

All the individuals have support values. An individual is fittest for the frequent area of a lexicographic tree whether the support value of this individual is greater or equal to the user defined threshold value. If the support value of an individual is greater than or equal to min_supp, then the fitness function will return a positive value for this individual otherwise it will return a negative value i.e.

If *support (individual)*≥min_supp **then** *fitness (individual)* = +1

Else *fitness (individual)* = -1.

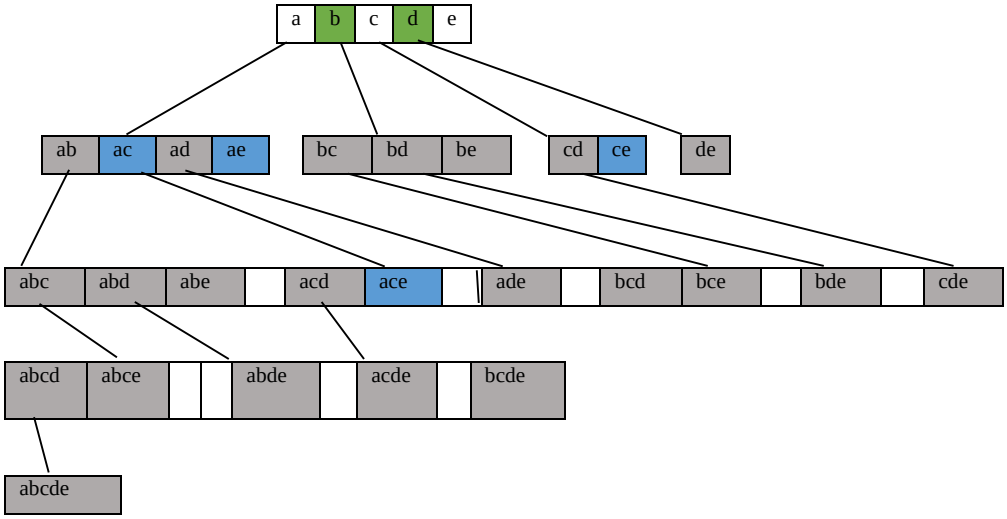


Figure 15: Illustration of the Hybrid GeneticMax approach to find maximal frequent item sets. Initially it sorted out infrequent items {b,d} from 1 –item sets {a},{b},{c},{d},{e} which is shown by green box. When it identify infrequent items then all the super item sets of these items will be invalidate chromosomes which are shown by black box. After then GeneticMax will apply to find maximal frequent item sets from item sets {ac}, {ae}, {ce} and {ace} which is shown by blue box.

4.3.2.7 Item Set Enumeration

If a data set contains long item sets, it generates huge candidate item sets which finally reduces the efficiency of a solution. A long item set enumerates combinatorial number of shorter, frequent sub item sets. For example, if a data set contains 50 item sets, such as $\{i_1, i_2, i_3, \ldots, i_{50}\}$, which enumerate $\binom{50}{1}$ frequent 1-itemsets:$(i_1, i_2, \ldots, i_{50})$, $\binom{50}{2}$ frequent 2-itemsets:(i_1, i_2), (i_1, i_3), … , (i_1, i_{50}), (i_2, i_3), (i_2, i_4),…, (i_2, i_{50}) and so on.

If the length of an item set is n, then it enumerates 2^n-1 frequent sub-item sets.

This is too large for computation and storage if the length of an item set is long. For each sub item set, the Apriori algorithm needs to be used to scan the data sets and calculate the support value of that item set. This increases the computational time of the algorithm and decreases its efficiency (Kabir et al. 2015b). This algorithm is acceptable if the position of the solution is near to the root in a lexicographic tree. On other hand, if the position of the solution is far from the root then it needs to consider huge amount of candidate item sets to reach the solution nodes.

For this reason, computational time increases as it considers larger amounts of candidate item sets, especially if the position of the solution is far from the root. To overcome this low efficiency of the Apriori algorithm, GeneticMax uses a global search mechanism which starts from any position of the lexicographic tree (Kabir et al. 2014; Kabir et al. 2015b). If the generated individual is infrequent, then all of its super item sets are infrequent and those item sets are automatically pruned. Similarly if the generated individual is frequent then all of its sub item sets are frequent and those item sets are automatically pruned. Through this technique, the search space of GeneticMax algorithm becomes narrower and narrower. Hybrid GeneticMax improves the GeneticMax algorithm by introducing local search. Initially, the infrequent items are sorted out from 1- item sets. The effect of sorting out the infrequent items at initial stage is shown in Figure 15.

4.4 Association Rule Mining for Both Frequent and Infrequent Items Using PSO

Particle swarm optimization (PSO) algorithm is an evolutionary computational technique where swarm describes the behaviour of particles. It was first introduced by Kennedy and Eberhart in 1995 (Eberhart & Shi 2001). To get the optimum solution, this technique considers a population based searching mechanism where particles change

their positions in a given space with respect to time. The particles are flying in a multi-dimensional space to find the solution in a PSO system. When particles fly in a multi-dimensional space, each particle considers two experiences to modify its current position. One is the best fitness value it has achieved called "pbest" and another is the best fitness value achieved by any particle of the generated population called "gbest". If $v_i(t)$ is the velocity of *i-th* particle at time *t*, then for calculating the new velocity of *i*-th particle at time *t+1*, which considers two best values pbest and gbest and it is

$$v_i(t+1) = v_i(t)+r_1(pbest_i - x_i(t))+r_2(gbest - x_i(t)) \quad (5)$$
$$x_i(t+1) = x_i(t)+v_i(t+1) \quad (6)$$

Where $x_i(t)$ is the position of particle *i* at time *t*, $pbest_i$ is the personal best position found by the *i-th* particle. To balance global and local search, Shi and Eberhart in 1998 introduced another method named "inertia weight". In this method the following equations are used to modify the position of a particle *I* (Shi & Eberhart 1998).

$$v_i(t+1) = wv_i(t)+r_1(pbest_i - x_i(t))+r_2(gbest - x_i(t)) \quad (7)$$

Here w acts as an inertia weight which can be a constant or time function. A constriction factor was added to the PSO technique by Clerc et al in 1999 (Clerc 1999). This factor increases the social interaction among the particles which is a major factor for improving the performance of the PSO algorithm.

4.4.1 Population Generation

The evolution starts with randomly generated individuals called particles. Each particle class contains 5 variables named support: support value of the particle; velocity: velocity of that particle; best_support; position: particle's current position; bestposition: particle's best position achieved so far, it is equivalent to "pbest". Here position is equivalent to the item sets in a lexicographic tree. At time, *t*, each particle's support value is compared with all other's, the best particle is chosen, where the support value of that particle is closest to the user defined support value. The support value of a best particle is called here 'gbest'. All other particles change their position with respect to the best particle's position. When the particles are traversed in a tree, all the traversed nodes are classified into frequent and infrequent item sets. In this way the search space is narrowed for the swarm particles.

4.4.2 Lifetime of Proposed Method

The lifetime of the proposed method depends on the fixing of all the particles position. The search space become so narrow that the swarm particles do not get the new solution after a certain amount of time and the program is terminated.

4.4.3 Algorithm for ARM Using PSO

According to the above description, the PSO algorithm for mining association rules for both frequent and infrequent items is summarized in the following flowchart:

Input D: database; numberofParticles; min_supp: minimum support; min_conf: minimum confidence;
Output AR: Association Rules

1) Scan the database D and find the support value of all 1- itemsets and store it in a matrix named Itemset_1_support
2) Prune those 1-itemset which do not satisfy min_supp threshold
3) Let *freq* be a matrix which store all frequent 1-itemsets
4) For each 1-itemset from matrix itemsets_1_support set IR_front and calculate IR_remaining = number_of_items – IR_front, this will finally give a search space for specific 1-itemsets
5) Generate random positions of the particles where the positions are in specific 1-itemset range search space
6) Check the FI_Superset_Member and NFI_Subset_Member array for superset and subset checking of these generated particles.
7) If any superset or subset of a particle is found in FI_Superset_Member or NFI_Subset_Member respectively, then this particle position is assigned as an invalid position and go to Step 4.
8) All the support values of particles are considered and compared it to the user defined threshold support value name min_supp. Support value of a particle which is close to the min_supp is assigned as best_particle.
9) If best_particle.support < min_support
   Since the itemsets closer to the root are more frequent, so the search space of that particle is above the current position in lexicographic tree. Perform NFI_Member_Add, and if any infrequent itemsets are found then update NFI_Subset_ Member.
10) If best_particle.support > min_support
    Since the itemsets far from the root are less frequent, so the search space of that particle is below the current position in lexicographic tree. Perform FI_Member_Add, and if any frequent itemsets are found then update FI_Superset_Member.

// The search space for swarm particles become narrows through step 9 and 10 and all the solutions would be near the cut which was shown in a lexicographic tree

11) All other particles follow the position of best particles and change their position randomly to avoid local optima.
12) For each frequent k-itemset $I \in FI\_Superset\_Member$
    If $c \geq \min\_conf$ generate association rules for this itemset

4.5 Extracting Interesting Rules Using ARMGAAM

4.5.1 Basic Concepts and Definitions

This section recalls the basic concepts and definitions of association rule mining and genetic algorithm.

4.5.1.1 Association Rule Mining

Initially, association rules were used in market-basket analysis but its application has extended to different real world fields including e-commerce, telecommunication, intrusion detection, bioinformatics, web mining. (Han & Kamber 2006, pp. 21-39).

Let $I = \{i_1, i_2, \dots, i_r, .., i_{n-1}, i_n\}$ is an item set which contains *n-numbers* of items and the transaction database is $T = \{t_1, t_2, \dots, t_{k-1}, t_k\}$ which contains *k-numbers* of transactions. Each transaction of *T*, i.e. t_i where $i \in k$, is a subset of an item set, *I*, such that $t_i \subseteq I$. If *A* and *B* are two item sets, then an association rule between these item sets is defined by $A \rightarrow B$, where $A \subseteq I, B \subseteq I$ and $A \cap B = \phi$. An association rule, $A \rightarrow B$, where *A* is antecedent and *B* is its consequent. Support and confidence are the two quality measurement factors for evaluating the validity of an association rule *A→B*, which are defined as follows:

1) The support value of a rule *A→B* is defined by the term, $supp(A \cup B) = \frac{|(A \cup B)(T)|}{|T|}$.

2) The confidence value of a rule *A→B* is defined by the term, $conf(A \rightarrow B) = \frac{supp(A \cup B)}{supp(A)}$.

That is, support means the occurring frequency of an item set in a database and strength of a rule is measured by confidence. Users provide the two threshold values of minimum support, min_supp and minimum confidence, min_conf. If $supp(A) \neq 0 \; and \; supp(B) \neq 0$, then a rule *A→B* is valid if $supp(A \cup B) \geq \text{min\_}supp \; and \; conf(A \rightarrow B) \geq \text{min\_}conf$. This support-confidence constraint is given by (Agrawal et al. 1993).

The association rule mining task is exemplified by the following example of a video shop database. There are different types of video cartoon series in the video shop. In this example, the following five video cartoon series are considered. There are eight transactions in the database concerning who bought those cartoon series.

Name of the Items	Item Code
Tom and Jerry	a
Meena Cartoon	b
Looney Tunes	c
The Flintstones	d
Scooby-Doo, Where are you!	e

a)Transactions	
1:	{a,b,d}
2:	{c,d,e}
3:	{a,c,d}
4:	{a,b,c,e}
5:	{b,c,e}
6:	{b,c,d}
7:	{d,e}
8:	{a,b,d,e}

b) Frequent item sets (min_supp = 2)			
0 item	1 item	2 items	3 items
{}: 8	{a}: 4	{a,b}: 3	{a,b,d}: 2
	{b}: 5	{a,c}: 2	{a,b,e}: 2
	{c}: 5	{a,d}: 3	{b,c,e}: 2
	{d}: 6	{a,e}: 2	
	{e}: 5	{b,c}: 3	
		{b,d}: 3	
		{b,e}: 3	
		{c,d}: 3	
		{c,e}: 3	
		{d,e}: 3	

c) Maximal frequent item sets
{a,b,d}, {a,b,e}, {b,c,e}

Figure 16: a) A simple transaction database of 8 transactions containing 5 items, b) Frequent item sets based on a user defined threshold value, min_supp = 2 (25%) and c) maximal frequent item sets based on table b).

From eight transactions and five items, three maximal frequent item sets are generated, which are shown in Figure 16. An item set is maximal, if there are no frequent supersets of this item set. These generated maximal frequent item sets satisfy the user defined threshold value which is 25% for this example. For every maximal frequent item set X, let $B \subseteq X \; and \; A = X - B$.

Association rules: min_conf = 30%		
$a \rightarrow b,d$ (conf: 50%, supp(a): 50%, supp(b,d): 38%, valid)	$b \rightarrow c,e$ (conf: 40%, supp(b): 63%, supp(c,e): 38%, valid)	$a \rightarrow b,e$ (conf: 50%, supp(a): 50%, supp(b,e): 38%, valid)
$a,b \rightarrow d$ (conf: 67%, supp(a,b): 38%, supp(d): 75%, valid)	$b,c \rightarrow e$ (conf: 67%, supp(b,c): 38%, supp(e): 63%, valid)	$a,b \rightarrow e$ (conf: 67%, supp(a,b): 38%, supp(e): 63%, valid)
$b \rightarrow a,d$ (conf: 40%, supp(b): 63%, supp(a,d): 38%, valid)	$c \rightarrow b,e$ (conf: 40%, supp(c): 63%, supp(b,e): 38%, valid)	$b \rightarrow a,e$ (conf: 40%, supp(b): 63%, supp(a,e): 25%, valid)
$d \rightarrow a,b$ (conf: 34%, supp(d): 75%, supp(a,b): 38%, valid)	$e \rightarrow c,b$ (conf: 40%, supp(e): 63%, supp(c,b): 38%, valid)	$e \rightarrow a,b$ (conf: 40%, supp(e): 63%, supp(a,b): 38%, valid)
$a,d \rightarrow b$ (conf: 67%, supp(a,d): 38%, supp(b): 63%, valid)	$b,e \rightarrow c$ (conf: 67%, supp(b,e): 38%, supp(c): 63%, valid)	$a,e \rightarrow b$ (conf: 100%, supp(a,e): 25%, supp(b): 63%, valid)

Figure 17: Few association rules which are generated from Figure 16 c)

A rule $A \rightarrow B$ is valid, if it satisfies a user defined confidence value, i.e. $conf(A \rightarrow B) \geq \min\_conf$. Association rules are generated from these maximal frequent item sets of Figure 16c), which satisfy a user defined minimum confidence value of 30% for this

instance. Association rules based on the support-confidence framework are shown in Figure 17.

4.5.1.2 Problem Statement

From the above discussion it can be seen that, the first step of extracting an association rule is to generate frequent item sets. The following major challenges are introduced by support-confidence dependent mining algorithms:

1) Extracting frequent item sets which satisfy a minimum support value reveal an exponential search space of 2^n, where n is the number of item sets. The last step considers generating all rules from frequent patterns having a minimum confidence value. The complexity of generating all rules is $O(k.2^r)$, where k is the number of frequent item sets and r is the length of longest frequent item set.

2) Since the support-confidence framework heavily depends on a user defined minimum support value, the performance of a mining task depends on the specification of a suitable threshold value by the user. If a user sets a big value as a threshold, no frequent item set is generated and a user will miss interesting patterns. Whereas setting a small threshold value will lead to lower performance due to generating a large number of frequent item sets. Moreover, a large number of different patterns are generated for different support levels and those are not interesting to the users.

3) Mining association rules from different databases require different support values. For example, the support values of item sets of TicTacToe and Zoo databases are distributed in the interval [0, 0.47] and [0, 0.82] respectively. TicTacToe database contains 958 instances, whereas 101 records are contained by Zoo. For Tictactoe database, no item sets are found if users set min_supp = 0.6 because it crosses the interval limit of this database. If a user selects same threshold value for a Zoo database, few interesting item sets are generated because the support value is in between the interval.

4) Apart from the above mentioned problems, support dependent mining algorithms introduce another major issue. From Figure 17, rule $d \rightarrow a, b$ [conf: 34%, supp(a,b): 38%]is discovered as a valid rule since it satisfies the minimum confidence value. However, this rule is misleading since the purchasing of the cartoon series (a,b) is 38% which is larger than the confidence value, 34%. In fact, the item set in antecedent is

negatively correlated with the item sets in consequent since the buying of one of these items actually decreases the probability of purchasing the other.

These challenges demonstrate the difficulty for users to assign a suitable support value. This provided the motivation to design an evolutionary algorithm based mining algorithm which does not depend on a user defined support value.

4.5.2 ARMGAAM Algorithm

Prior to proceeding with the algorithm, the database needs to be processed. Let $I = \{I_1, I_2, \dots, I_{n-1}, I_n\}$ be a set of items and $D = \{A_1, A_2, \dots, A_{k-1}, A_k\}$ be a database which contains *k* number of attributes. Each attribute is classified into different type i.e. $A_i = \{t_1, t_2, \dots, t_{j-1}, t_j\}$. Each type of an attribute is termed as an item, i.e. $A_i = \{I_1, I_2, \dots, I_{j-1}, I_j\}, A_{i+1} = \{I_{j+1}, I_{j+2}, \dots, I_k\}$ and so on. For example, in mushroom database, the attributes cap-shape and cap-surface are classified into {bell, canonical, convex, flat, knobbed, sunken} and {fibrous, grooves, scaly, smooth} respectively. These two attributes are transformed into set of items $I = \{I_1 = bell, I_2 = canonical, I_3 = convex, I_4 = flat, I_5 = knobbed, I_6 = sunken, I_7 = fibrous, I_8 = grooves, I_9 = scaly, I_{10} = smooth\}$.

This section describes the proposed model named Association Rules Mining with Genetic Algorithm Using an Adaptive Mutation Method (ARMGAAM), for mining a reduced set of BARs with a good tradeoff between the number of generated rules and good coverage of the data set, considering three objectives: lift, net confidence and conditional probability, including support and confidence values. In order to perform an evolutionary learning, this approach extends the ARMGA algorithm and introduces two new components: the reinitialization process and an adaptive mutation method to its evolutionary model. In the following, all the characteristics are briefly described (see Section 4.5.2.1-4.5.2.5) and a flowchart of the algorithm is represented (see Section 4.5.2.6).

4.5.2.1 Objectives

The basic concepts of a traditional association rule mining task is to find all rules where the support and confidence values of those rules are larger or equal to the user defined minimum support, min_supp and a minimum confidence, min_conf, respectively.

In recent years, several authors have proposed different measures according to the potential interest of the users (Martin et al. 2014; Geng & Hamilton 2006). Some of those that are used in the current literature for mining BARs will be briefly explained.

The conditional probability measure of a rule analyses the dependence between *A* and *B* and it is defined as:

$$CP(A|B) = \{supp(A \cup B)\text{-}supp(A)supp(B)\}/\{supp(A)(1\text{-}supp(B))\} \quad (8)$$

Its obtain values in [- ∞, ∞], where misleading rules are represented by 0 > value > -∞, 0 < value < ∞ represents positive association rules, and value = 0/–∞/∞ represents trivial rules. The ratio between the confidence and the expected confidence of a rule is measured by lift and it is defined as,

$$lift(A \rightarrow B) = supp(A \cup B) / \{ supp(A)supp(B)\} \quad (9)$$

The netconf measure is used to evaluate a rule based on the support value of that rule and its consequent and antecedent support. Its domain range is [-1,1], where positive values, negative values and zero represent positive dependence, negative dependence and independence, respectively. Netconf of a rule $A \rightarrow B$ is defined as follows:

$$netconf(A \rightarrow B) = [supp(A \cup B)\text{-}\{supp(A)supp(B)\}]/[supp(A)(1\text{-}supp(A))] \quad (10)$$

Three objectives: lift, net confidence and conditional probability are maximized in order to get interesting knowledge from a data set. In this study, strong rules are generated which show a strong dependence among the item sets and avoid the problem of the support-confidence framework. Notice that, positive association rules represent positive dependence, thus this algorithm is focused on those rules that have CP>0. It is important to note that existing ARMGA generates those BARs which have higher CP value but low values in other objectives. This approach is focusing on those BARs which maintain a good trade-off between the number of generated rules and in all other measures of a data set.

4.5.2.2 Encoding

Like traditional ARMGA, the proposed model also follows the Michigan strategy (Yan et al. 2009; Yan et al. 2005) to encode each association rules into a single chromosome. Given an association rules of *k* length means that, a rule contains *k* items. Figure 18 shows the configuration of the chromosome which contains *k-genes*, i.e. *k* number of items. The first place of the chromosome contains a number which acts as an indicator for separation of antecedent from the consequent. So, the *k-rule* represents a chromosome of length k+1. For example a rule is $A \rightarrow B$, where antecedent *A* contains $item_1$ to $item_n$ and consequent *B* contains $item_{n+1}$ to $item_k$, where $0 < n < k$. The presence of an item in a chromosome represents by "1", otherwise "0" means the absence of that item in a chromosome.

n	$item_1$	$item_2$	…	$item_n$	$item_{n+1}$	…	$item_k$

Figure 18: A chromosome of an association rule of *k* length

4.5.2.3 Initialization of Population

Given a rule length *k*, a seed chromosome and a population size, a random function is used to initialize a population.

```
population initialization (seed_chromosome, sizeof_population)
Begin
  pop[0] ← seed_chromosome;
  i ← 0
  while i ≤ sizeof_population
  Begin
      for ∀c ∈ pop[0] do
      Begin
          c.item0 ← rand(k-1)+1;
          for j← 0 to k
          Begin
              c.itemj ← rand(number_of_items) + 1;
          End
      End
  End
  return pop[0];
End
```

Figure 19: An algorithm for initialization of the population

An algorithm for initializing the population is shown in Figure 19. Sizeof_population is a constant which defines the maximum number of chromosomes in a population. Total number of items presented in a database is defined by the variable name num-

ber_of_items. Here, function rand (number_of_items) is a random function which generates random number from 0 to number_of_items. If the generated number is *x*, where $x \in [1, number\_of\_items]$ in a chromosome, this represents $item_x$ is present in the chromosome. To maintain the uniqueness of all the positions in a chromosome, a generated random number for a specific item position in a chromosome is checked with other item positions in that chromosome. For a specific position, if the generated number is already used by another item position of a chromosome, then a new random number is generated for that position until it is a unique number.

4.5.2.4 Genetic Operators

Three genetic operators, selection, crossover and adaptive mutation, are designed for this proposed approach. These operators are discussed in this subsection.

Selection

By using this operator, an individual chromosome is chosen from a given population. This operator acts as a filter to choose an individual chromosome based on the fitness function and selection probability (*sp*). The value of a probability, *sp*, is set high to explore more on the search space. The select function select(*chrom*, *sp*) returns TRUE,

1) if the fitness value of a given chromosome is higher than the probability, *sp*, or
2) if the multiplication result of a random value generated by a random function with the fitness value of a given chromosome is less than the probability *sp*.

If a given chromosome fails to satisfy the above criteria then the select function returns FALSE.

```
Boolean select (chrom, sp)
  Begin
  if (fitness(chrom) > sp) then
     return TRUE
  Else
     if (fitness(chrom)*random_func() < sp)
        return TRUE
     Else
        return FALSE
```

random_func() will generate a random value ranged from 0 to 1, fitness (*chrom*) is the fitness function of ARMGAAM model which will returns the fitness value of a given chromosome.

Crossover

Crossover is one of the significant features of genetic algorithms. This operator is applied on two chromosomes of a given population called parent chromosomes to reproduce two new offspring chromosomes by exchanging parts of the parent chromosomes.

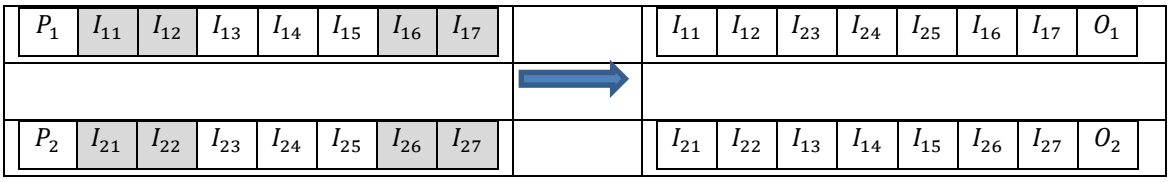


Figure 20: Two-point crossover example

ARMGAAM model uses two-point crossover mechanism where the two crossover points are generated randomly. That is any segment of parent chromosomes are chosen. The two-point crossover operation is illustrated by Figure 20.

Adaptive Mutation

This approach extends the existing ARMGA algorithm, which uses a conditional probability as a fitness function and a genetic algorithm to optimize the association rule mining problem. In this study, ARMGAAM uses an adaptive mutation method (Kannimuthu & Premalatha 2014). The mutation operator is used to keep the diversity from one generation of a population to the next one. Mutation changes one or more genes of a chromosome with respect to a mutation probability, *mp*. Existing ARMGA follows fixed mutation probability for mutation operation. Normally, the value of mutation probability (*mp*) is set to low. The search will become a random search if it is set too high. An adaptive mutation based approach provides better performance than fixed mutation (Kannimuthu & Premalatha 2014). In the proposed approach, the rate of mutation is reduced with the increasing of the generation number. Mutation rate is adapted with respect to the fitness value of the offspring. Initially, a large mutation rate is applied for more exploration on the search space (to explore more on the search space). Based on the fitness value, the offspring are categorized into different ranks. The rate of mutation is assigned for an offspring based on its rank. Top ranked offspring are mutated at a lower rate than lower ranked ones.

4.5.2.5 Reinitialization Process

The reinitialization process is used to move away from local optima and increase the diversity in the population. This process is only applied if the number of new chromosomes of a population is less than α% of a current population.

4.5.2.6 Extracting Positive Association Rules with Potential Interest using Genetic Algorithm

A method for mining association rules which efficiently extracts interesting positive association rules from the database using a genetic algorithm without specifying a user defined threshold value is proposed. Many frequent item sets of positive rules are not interesting and some of the interesting rules are missed in the intermediate generation of a population because of using a fixed mutation approach.

PROCEDURE. ARMGAAM
Input D: Dataset D, seed_chromosome, sizeof_population, selection probability sp, crossover probability cp, rule length k
Output: Positive association rules with potential interest
(0) Categorical attributes of a data set are mapped into Boolean attributes
(1) **begin** i ← 0
(2) pop[i] ← initialization (seed_chromosome, sizeof_population)
(3) **while** not terminate_func(pop[i]) **do**
(4) **begin** temp_pop_after_selection[0] ← 0
(5) temp_pop_after_crossover [0] ← 0
(6) **for** ∀$chrom$ ∈ pop[i] **do**
(7) **if** select(chrom, sp) **then**
(8) temp_pop_after_selection[0] ← chrom
(9) temp_pop_after_crossover[0]←crossover(temp_pop_after_selection, cp)
(10) **for** ∀$chrom$ ∈ temp_pop_after_crossover[0] **do**
(11) pop[i] ← rank_based_adaptive_mutation (chrom)
(12) i ← i+1
(13) If the number of new chromosomes in a current population is less than α% of the previous population, then reinitialize the population.
(14) end
(15) **return** pop [i]
(16) end

Figure 21: Procedure of ARMGAAM

The search space is significantly reduced if the extracted frequent item sets are restricted to positive rules of potential interest. For this reason, an efficient algorithm is designed based on GA using an adaptive mutation method which generates positive association rules of potential interest. At an earlier stage, the attributes of a database are pro-

cessed and transformed into Boolean item sets. According to the above description, the proposed approach for mining BARs can be summarized through Figure 21.

In this algorithm, an initialization function is used to initialize the current population pop[*i*]. select operator is applied on the current population to choose chromosomes based on a select probability, *sp*. These chromosomes are stored in an array called temp_pop_after_selection. Then pairs of chromosomes of temp_pop_after_selection are crossed over based on a crossover probability, *pc* to reproduce two new offspring and stored in temp_pop_after_crossover array. This process is continued until all the chromosomes of temp_pop_after_selection are crossed over. Each chromosome of temp_pop_after_crossover array is mutated using adaptive_mutation method. This proposed approach is stopped if the maximum number of evaluation is reached.

4.6 Extracting Interesting Rules Using MBAREA

4.6.1 Objectives

The classical algorithms focus on generating association rules if these satisfy user defined support and confidence values. Most of these algorithms focused on high support based rules, that is those rules which frequently appeared in a data set (Jesus et al. 2011; Martin et al. 2014). However several authors have noticed some drawbacks of this structure which leads to the generation of a huge number of misleading and trivial rules (Qodmanan et al. 2011; Martin et al. 2014). A rule is misleading if supp(Y) > confidence ($X \rightarrow Y$) i.e. the item set in the antecedent is negatively correlated with the item sets in the consequent, since the buying of one of these items actually decreases the probability of purchasing the other (Zhou & Yau 2007).

Many researchers used different measures for evaluating the quality of a rule and those approaches significantly reduced the generation of misleading rules (Mukhopadhyay et al. 2014; Qodmanan et al. 2011; Yan et al. 2009). Those approaches still suffered from generating misleading as well as trivial rules. A rule $X \rightarrow Y$ is trivial, if supp(X) = 0 or supp(Y) = 0, then supp($X \cup Y$) = 0 for any item sets of X or Y, respectively.

In the previous section 4.5.2.1, different measures have been discussed. Along with those measures, a new measure, named interestingness, is used for this approach to evaluate the quality of a rule.

For finding interesting rules, new rules are generated based on each item present in the consequent part of a rule. Since a number of items are present in the consequent part of a rule and it is not predefined, this approach may not be suitable for an association rule mining task. Recall the definition of interesting (Wakabi-Waiswa & Baryamureeba 2008), a new expression for measuring the interestingness of a rule $A \rightarrow B$ is defined as follows:

$$I = [\text{supp}(A \cup B)/\text{supp}(A)] \times [\text{supp}(A \cup B)/\text{supp}(B)] \times [\text{supp}(A \cup B)/|D|] \quad (11)$$

Here, I is the interestingness constraint of a rule $A \rightarrow B$ and the total number of records in a database is defined by the term |D|. Its domain range is [0, ∞], where 0, ∞ and 0 < value < ∞ represents independence, trivial rules and positive dependence, respectively.

Three objectives are maximized for this problem: conditional probability (*CP*), lift and interestingness. This study is only interested in mining very strong rules which have positive dependence between items, avoiding the problem of support-confidence framework based methods. Notice that positive association rules allow to represent positive dependence. Thus, a rule $X \rightarrow Y$ must satisfy the following conditions: i) $CP > 0$; ii) $\text{supp}(X \cup Y) > 0$; iii) $\text{supp}(X) \neq 0$ and $\text{supp}(Y) \neq 0$. In this study, *CP* will act as a fitness function of a valid rule for filtering out misleading and trivial rules. A rule with a *CP* value near to one means a high degree of positive dependence between item sets and may be more important to the users. Interestingness is a measure of a rule through which one can say how interesting a rule to the users. Here the well-known interestingness measure is used. Since its range is not bounded, the better value denotes the difference between the rules and reduces the number of generation.

4.6.2 Genetic Operators

A chromosome is a gene vector which represents the attributes and an indicator for separation between item sets. Given an association rule of k length means that, a rule contains k items which is shown by Figure 22.

n	$item_1$	$item_2$	…	$item_n$	$item_{n+1}$	…	$item_k$

Figure 22: A chromosome of an association rule of *k* length

For example a rule is $A \rightarrow B$, where antecedent A contains $item_1$ to $item_n$ and the consequent B contains $item_{n+1}$ to $item_k$, where $0 < n < k$. The first place of a chromosome is an indicator for separation from antecedent to consequent.

By using selection operator, an individual chromosome is chosen from a given population. This operator acts as a filter to choose an individual chromosome based on the fitness function and selection probability (*sp*). Crossover is one of the significant features of genetic algorithms. This operator is applied on two chromosomes of a given population called parent chromosomes to reproduce two new offspring chromosomes by exchanging parts of the parent chromosomes. A two-point crossover mechanism is used by MBAREA where the two crossover points are generated randomly. That is, any segment of parent chromosomes is chosen. Mutation operator is explained in the following section.

4.6.3 Class Based Mutation and Best Population

In order to store all the non-dominated rules which are generated in the intermediate generation of a population, provoking the diversity of the population, and increasing the coverage of data sets in this study a class based mutation approach along with best population method are introduced. The mutation operator is used to keep the diversity from one generation of a population to the next one. Mutation changes one or more genes of a chromosome with respect to a mutation probability, *mp*. Existing GA based approaches such as ARMGA and ARMMGA, followed fixed mutation probability and randomly mutated the chromosomes. Although these methods used low mutation probability, it mutated few high quality chromosomes due to the random function. This resulted in some top quality chromosomes having less chance to contribute to future generations of a population. To prevent this problem and to give more chance to the best chromosomes to contribute to future generations, the proposed approach classifies the whole population into δ, based on a fitness value of each chromosome. Top class chromosomes have a higher fitness value but assign with a low mutation ratio whereas low class chromosomes are mutated with high mutation probability. Through this approach high class chromosomes take part for future generations of a population.

Best population (*BP*) keeps all the non-dominated rules which are generated in intermediate generations of a population. Moreover, *BP* is updated with the generation of a new

population following the non-dominance criteria. This process helps us to increase the coverage of a data set and for performing enhanced exploration of the search space.

4.6.4 MBAREA Algorithm

According to the above description, the MBAREA algorithm is summarized through the following structure.

PROCEDURE: **MBAREA**	
Input:	Database D, size of population popSize, selection probability sp, crossover probability cp, class_size δ, mutation probability mp[δ], rule length k
Output:	Best Population (Association rules with potential interest)
(0)	categorized attributes of a database D, are mapped into Boolean attributes
(1)	i ← 0, bestPopulation[i] ← 0, mp[δ] ← mutation_function(δ)
(2)	population[i] ← initialization (popSize)
(3)	**while** not termination_condition_reached (population[i])
(4)	pop_after_selection[i] ← 0
(5)	pop_after_crossover[i] ← 0
(6)	**for** ∀chromosome ∈ population[i] do
(7)	pop_after_selection[i] ← selection(population[i], sp)
(8)	pop_after_selection[i]←sort_population(pop_after_selection[i])
(9)	pop_after_crossover[i] ← crossover (pop_after_selection[i], cp)
(10)	pop_after_crossover[i]←sort_population(pop_after_crossover[i])
(11)	population[i] ← class_based_mutation (pop_after_crossover[i], mp[δ])
(12)	best_population ← non-dominated_rules(population[i])
(13)	remove redundant rules from best_population
(14)	i ← i+1
(15)	**End**
(16)	**The best_population is returned.**

Figure 23: Procedure of MBAREA

This process continues until any of the following conditions are met:

1) the maximum number of evaluation is reached, or
2) the average value of the fitness function of the current population is less than the value α of a previous population.

4.7 Multiple Seeds Based Evolutionary Approach for Mining Association Rules

In this section all the characteristics of multiple seeds based genetic algorithm are explained and finally the pseudo code of MSGA algorithm is described in a flowchart.

4.7.1 Distance Measure and Multiple Archive Design

In different types of combinatorial problems, distance measures are strongly problems dependent. There are two ways to measure the individual distance in combinatorial problems:

1) a measurement of the difference between two chromosomes, while the other is
2) a structural difference measure based on a mathematical foundation.

Suppose *X* and *Y* are two chromosomes and the length of each of those chromosomes is denoted by *L*. There are two different methods which are used to measure the distance between two genomes and those are listed as follows:

4.7.1.1 Hamming Distance Method

The hamming distance method is used to measure the distance between two chromosomes of equal length by the number of positions where the corresponding symbols are different. In other words, it measures the minimum number of substitutions which are required to change one chromosome into the other one. If *I* is an indicate function, then the hamming distance between two chromosomes *X* and *Y* is defined by the following equation:

$D(X,Y) = I/L$, where, $I = \sum_{j=0}^{L} I_j$, $I_j = \begin{cases} x_j = y_j, 0 \\ x_j \neq y_j, 1 \end{cases}$ (12)

4.7.1.2 Euclidean Distance Method

This method is used for real encoding. Like hamming distance, this method is used for measuring the distance between two chromosomes. In general, for an *L* dimensional space the distance between two chromosomes is:

$$D(X,Y) = \sqrt{\sum_{i=1}^{L} (x_i - y_i)^2} \quad (13)$$

In this study, real encoding technique is used, i.e. Euclidean distance method to measure the distance between two chromosomes. The convergence of a population during the evolutionary learning is described by the following distance measure:

$$d(P) = \frac{1}{n}\sum_{i=0}^{n-1}\sum_{j=1}^{n-1}(D(I_i, I_j))^2 \quad (14)$$

Where *n* is the size of a population, I_i and I_j are *i-th* and *j-th* individual, and *D* is the distance between two individuals.

To find the high quality association rules from a large solution space, this approach subdivides the whole solution space into m-domains. In the following, the encoding technique of an association rule, subdividing the whole solution space into m-domains, chromosome generation from each domain, selecting seeds from domains, and generating an initial population from multiple seeds are briefly described.

4.7.2 Encoding

The proposed model follows the Michigan strategy (Yan et al. 2005; Yan et al. 2009) of encoding each association rule into a single chromosome. Given an association rule of *k* length means that, a rule contains *k* items. Figure 24 shows the configuration of the chromosome which contains *k-genes*, i.e. *k* number of items. The first place of the chromosome contains a number which acts as an indicator for separation of the antecedent from the consequent. So, the *k-rule* represents a chromosome of length *k*+1. If a rule is *A→B,* where antecedent *A* contains $item_1$ to $item_n$ and the consequent *B* contains $item_{n+1}$ to $item_k$, where $0 < n < k$ and *item₁…item*$_k \in$ *set of items, I.*

n	$item_1$	$item_2$	…	$item_n$	$item_{n+1}$	…	$item_k$

Figure 24: A chromosome of an association rule of *k* length

For example, if *X* and *Y* are item sets which contain items from a set of items *I* of a data set *D*, then the configuration of an association rule *X→Y* can be represented as follows:

2	45	7	21

Figure 25: An example of a chromosome

Here the rule length *k* = 3, *n* = 2, *X* = {45,7}, *Y* = {21} and the number of items in a data set *D* is not less than 45.

4.7.3 Division of a Solution Space

For dividing the whole solution space into m-domains, this approach considers each chromosome as a position in an *L* dimensional solution space. To do this, it measures the distance between initial to end item sets, which is the maximum distance in a solution space. Each item in a data set *D* is assigned by a unique identifier. For example, if a data set contains *n* items then $item_1$, $item_2$,....,$item_n$ are assigned by *ID 1,2,....n*. In order to measure the distance between two chromosomes, the first position is ignored that is an indicator of a chromosome of an association rule of Figure 24. For an association rule of *k- length*, the initial and end chromosomes are defined by the following figures.

1	2	3	…	k

Figure 26: An initial chromosome

n-(k-1)	n-(k-2)	…	n-1	n

Figure 27: An end chromosome

Note that the order of the chromosome ID is important for defining the initial and end chromosomes. After calculating the Euclidean distance between an initial and end

0>range≤r	r>range≤2r	2r>range≤3r	….	(m-1)r>range≤ mr
$Domain_1$	$Domain_2$	$Domain_3$		$Domain_m$

Figure 28: Ranges of m-domains

chromosome, the next step is to equally divide the whole distance into *m-regions*. If the total distance is *d*, then the range (*r*) of each domain is calculated by the following equation, $r = d/m$. Based on the ranges, the whole solution space of a data set is subdivided into m equal size domains as shown in Figure 28.

4.7.4 Chromosome Generation from Each Domain

The following factors which are listed in Table 1 are the influential coefficients which are used for designing an archive.

These factors are described in this section along with the technique used for generating individuals. In the results and analysis chapter (chapter 5), the results are compared with different single seed based genetic algorithms. The cage size of an archive depends on the population size and the number of archives. Each domain is assigned an archive.

Table 1: Factors for designing an archive

Factors	
Policy for Archive population	Distance
Cage size of an archive	Population size/m
Population rank policy	fitness
Population size	varied

The member chromosomes of an archive are the members of the assigned domain. Whether a generated chromosome is the member of the assigned domain or not depends on the Euclidean distance between a generated and an initial chromosome. This approach randomly generates chromosomes and calculates

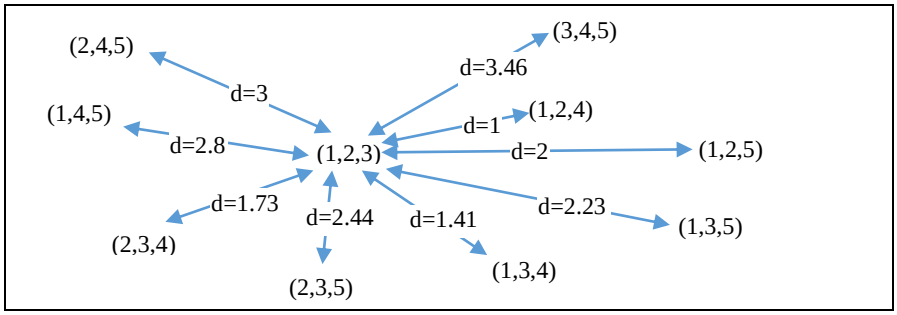


Figure 29: Distances of different chromosomes from an initial chromosome

the distance from the initial chromosome. Based on the distance and specified ranges of the domains (see Figure 28), it inserts the chromosomes to the specific archives. For example, if a data set has five items a, b, c, d, e and each item is assigned with an unique *ID*, i.e. {a=1, b=2, c=3, d=4, e=5}, the initial and final chromosome is (1, 2, 3) and (3, 4, 5), respectively. The Euclidean distance between initial and end chromosome is 3.46. If the four domains are considered, then the calculated range of each domain is, d/m = 3.46/4 = 0.865. Therefore, the four domains can be defined as, 0> $domain_1 \leq$ 0.865, 0.865 > $domain_2 \leq$ 1.73, 1.73 > $domain_3 \leq$ 2.595 and 2.595 > $domain_4 \leq$ 3.46. The distances of different generated chromosomes are shown in Figure 29. If the generated chromosome is (2, 3, 5), then the distance from initial chromosome is 2.44. Therefore this chromosome satisfies the range associated with $domain_3$, so this method add this chromosome to the archive which is assigned for this domain. Through this mechanism multiple archives are generated by gathering chromosomes from the assigned domain. Finally, the population of each archive is sorted according to the fitness value of each individual. As described in section 3.7, conditional probability is used as a fitness

function to define the quality of a rule. The fitness value of a rule is normalized between 0 and 1. The population of an archive is sorted with respect to its chromosome fitness values with *n/n…1/n* values assigned to each chromosome based on the fitness value. So the best chromosome of an archive obtains *n/n* that is assigned as a top ranked chromosome, whereas the worse chromosome of that archive gains *1/n*. Therefore, with this approach, a better chromosome of a domain has more chance to perform as a seed chromosome.

4.7.5 Generating an Initial Population from *m-seeds*

An effective initial population is generated from *m-seeds*. A top ranked chromosome from each archive is selected as a seed. Therefore, *m* archives generate *m* seeds. Note that, a seed cannot guarantee that this chromosome will have a high fitness value in a domain from where it was generated. Each seed uses a mutation function to produce an n individual, where the mutation probability is 1. Therefore, *m* seeds generate $m \times n$ individuals. These individuals are used as an initial population for the multiple seeds based genetic algorithm. The pseudo code of initialization function is shown below:

```
population_initialization (seed_chromosomes, population_size, no_of_archives)
(0)  begin i ← 0
(1)  for ∀c ∈ seed_chromosomes do
(2)    begin temp ← 0
(3)      while temp ≤ population_size/no_of_archives do
(4)          begin pop[i] ← mutate_for_initialization(c, 1)
(5)          i ← i+1
(6)          temp ← temp+1
(7)          end
(8)    end
(9)    return pop[i]
(10) end
```

In mutation for the initialization function, rand_func generates a random real number which has a range from 0 to 1, and rand_val(*k*) returns a random integer number ranged from 0 to *k*.

mutate_for_initialization (*c*, mutation_probability)	
(0)	**begin**
(1)	**if** (rand_func()×fitness(*c*) < mutation_probability) **then**
(2)	**begin** $c.item_0 \leftarrow$ rand_val(*k*-1)+1
(3)	i ← rand_val(*k*)+1
(4)	$c.item_i \leftarrow$ rand_val(no_of_item-1)
(5)	**end**
(6)	**return** *c*
(7)	**end**

4.7.6 MSGA Algorithm

According to the above description, the multiple seeds based genetic algorithm for discovering BARs is summarized through the following flowchart:

PROCEDURE: **MSGA**	
Input:	Data set *D*, population_size, no_of_archives, *sp* (selection probability), *cp* (crossover probability), *mp* (mutation probability), *k* (rule length)
Output:	A population consists of the positive association rules with conditional probability = 1
(0)	Mapping categorical attributes of a data set *D* into Boolean attributes
(1)	archives[no_of_archives] ← generate_archives(initial chrom, end chrom, no_of_archives)
(2)	seed_chromosomes[no_of_archives] ← seed_generation (archives)
(3)	**begin** i ← 0
(4)	pop[i] ← population initialization (seed_chromosomes, population_size, no_of_archives)
(5)	**while** not reach_generation_number (pop[i]) **do**
(6)	**begin** temp_pop[0] ← 0
(7)	**for** $\forall chromosome \in$ pop[i] **do**
(8)	**if** selection(chromosome, sp) **then**
(9)	temp_pop[0] ← chromosome
(10)	temp_pop[0] ← crossover(temp_pop[0], cp)

(11)	**for** $\forall chromosome \in$ temp_pop[0] **do**
(12)	temp_pop[0] ← mutation(chromosome, mp)
(13)	i ← i+1
(14)	**end**
(15)	**return** pop[i]
(16)	**end**

Figure 30: Procedure of MSGA

4.8 Chapter Summary

This chapter has presented all the characteristics of the proposed methods. This chapter also explained the underlying concepts and the structure of each approach along with the pseudo code for extracting frequent patterns. The basic concepts, objectives and the flowchart of the proposed algorithms for mining Boolean association rules were also discussed. Finally, the technique for encoding, generating an initial population from multiple seeds along with the pseudo code of multiple seeds based genetic algorithm was described.

In order to analyse the performance of the proposed approaches, the following chapter will discuss the experimental results.

Chapter 5 - Results and Analysis

5.1 Introduction

To show the effectiveness of the proposed algorithms which are justified and described in the previous chapters, the experimental analysis of these approaches are discussed in this chapter.

The performance of the GeneticMax algorithm for mining maximal frequent item sets is shown through the experimental results in section 5.2. The experimental results of the Hybrid GeneticMax algorithm is demonstrated in section 5.3 along with the comparative analysis of this method with GeneticMax algorithm. The experiments of the PSO based approach for mining association rules for both frequent and infrequent items are evaluated in section 5.4. The performance analysis of ARMGAAM and MBAREA algorithms for mining Boolean association rules is carried out on section 5.5 and 5.6, respectively. Finally, the experimental results of multiple seeds based genetic algorithm is discussed in section 5.7.

5.2 Mining Maximal Frequent Item Sets Using GeneticMax

5.2.1 Experimental Study

Several experiments are conducted on different real world data sets to analyze the performance of the proposed method. This section is organized as follows:

In subsection 5.2.2, a brief description of the algorithm is presented.

In subsection 5.2.3, the data sets which are used for these experiments are introduced.

In subsection 5.2.4, the proposed algorithm is evaluated.

In subsection 5.2.5, run time of the proposed algorithm is analyzed.

In subsection 5.2.6, the performance of the proposed method is compared with the most popular algorithm, Apriori (Agrawal & Srikant 1994; Hipp et al. 2000).

5.2.2 Experiments

The experiments are performed on an Intel(R) core i5-3210M CPU @2.50GHz, 4 GB RAM running on Windows 7 Enterprise. Microsoft Visual Studio 2012 is used to compile the code of GeneticMax. Three data sets including Tic Tac Toe, 10000×8 and Zoo are used to test GeneticMax. Different support values are applied to these data sets to check how many nodes are tested and the numbers of chromosomes are generated to get

the exact number of maximal frequent item sets, run times, and so on. Here, run time is the total execution time. GeneticMax embeds two main features: i) superset-subset relationship in both positive and negative boundaries in a lexicographic tree for pruning invalid chromosomes, and ii) use of a Genetic Algorithm which uses a global search mechanism. The purpose of this new approach is for convergence to a solution as fast as possible. A full experiment of GeneticMax on these data sets is conducted, demonstrating GeneticMax's ability to yield solutions rapidly by accessing the data sets for a few number of nodes in a lexicographic tree.

From the previous discussions it can be concluded that the Apriori algorithm tests all of the nodes in each level and prunes those nodes which do not satisfy a minimum support value. In GeneticMax, if it generates a chromosome X in any level which satisfies a minimum support value, then all the other subsets of X in any level are automatically pruned which dramatically reduces the time for accessing a large data set. This is also true the other way around: if GeneticMax generates a chromosome Y in any level which does not satisfy a minimum support value, then all the other supersets of Y in any level are automatically pruned.

5.2.3 Data Sets

The proposed approach is tested on different data sets such as Tic Tac Toe, Zoo, 10000×8 and so on. These data sets are taken from the University of California at Irvine (UCI) machine learning repository (http://archive.ics.uci.edu/ml/datasets.html) and University of Regina (http://www2.cs.uregina.ca/~dbd/cs831/notes/itemsets/datasets.php).

5.2.4 Evaluation of the Experiments

From the experimental results as shown in Table 2, it can be concluded that if the number of generations are increased then it increases the frequent item sets. For example, for the 10000×8 data sets, generation 100 produced 9 frequent item sets whereas generation 140 produced 8 frequent item sets. In other words, generation 100 resulted in more than 9 frequent item sets. On the other hand, generation 140 resulted in more than 8 frequent item sets. If these two generations are compared, it can be concluded that generation 100 still did not find some frequent item sets. When the number of generation are increased to 140, it found some item sets missed by generation 100. The same results are given by generation 140 and 150, respectively. So users can use generation 140 as a threshold value for the 10000×8 data sets. This is also true for TicTacToe. The same results are given

by generation 1200 and 1300, which contain the maximal frequent item sets. So for TicTacToe, user can use generation 1200 as a threshold value.

Table 2: The experimental results of GeneticMax for two different data sets

Database	Records	Items	Support (%)	Generation	Frequent Item sets	Time (s)	Remarks
10000×8	10000	8	20	100	9	10.22	
				140	8	21.67	This generation contains MFI
				150	8	25.10	
TicTac-Toe	958	9	16	100	6	10.13	
				250	7	17.528	
				500	10	43.83	
				1100	23	78.20	
				1200	24	95.60	Both Generations provide the same result
				1300	24	115.66	

The results in Table 3 show a comparison between the number of nodes in a lexicographic tree and the number of nodes which are tested for getting maximal frequent item sets. For 10000×8, there are 255 item sets and GeneticMax accessed only 39 item sets in the main data sets to get the maximal frequent item sets. Since GeneticMax uses the principles of genetic algorithm and prunes invalid chromosomes based on superset-subset relationship, it dramatically reduces the number of item sets out of a data set for obtaining the support value to mine maximal frequent item sets. The advantage of using those principles in GeneticMax is shown in Table 3, where (255-39) = 216 nodes are not examined to get the support value from data sets 10000×8 to get the exact number of maximal frequent item sets. Only 39 nodes are examined to get the final solution. For TicTacToe, only 114 nodes are examined to get the final solution (the other 397 nodes are not required).

Table 3: Results showing the number of times the data sets are accessed by GeneticMax

Database	Items	Support (%)	No. of nodes in the Lexicographic Tree (2^{items}-1)	No. of nodes tested for getting MFI
10000×8	8	20	255	39
TicTacToe	9	16	511	114
Zoo	17	50	131072	361

5.2.5 Run Time Analysis

As it can be seen from Figure 31, the runtime of GeneticMax increases with respect to

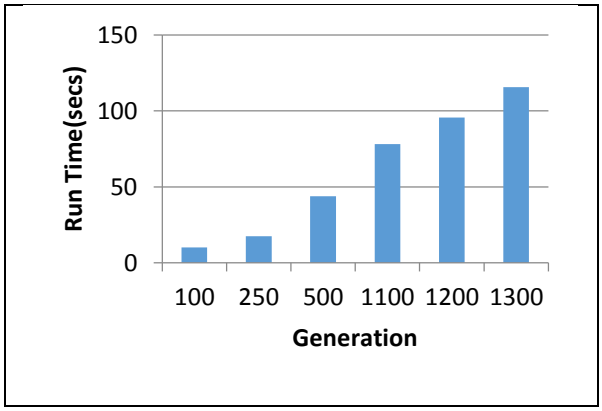


Figure 31: Run time versus Generation for TicTacToe

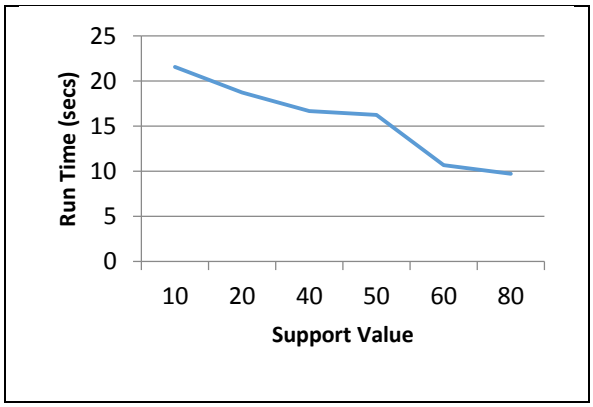


Figure 32: Run time of GeneticMax for different support values

the generation number of chromosomes. A lower support value which generates more frequent item sets needs higher runtime whereas a higher support value generating less frequent item sets needs less computational time, as shown in Figure 32.

5.2.6 Comparative Analysis of the Proposed Algorithm with Apriori

To verify the performance of the proposed GeneticMax method, a most popular algorithm named Apriori (Agrawal & Srikant 1994; Hipp et al. 2000) is used for finding maximal frequent item sets for a comparison study. Both of these algorithms are applied on the same data sets. C programming language is used for coding both of the algo-

rithms. The experiments are performed on an Intel(R) core i5-3210M CPU @2.50GHz, 4 GB RAM running on Windows 7 Enterprise. Microsoft Visual Studio 2012 is used to compile the code of the proposed method. The experiments are carried out on Real data sets as well as synthetic data sets (Agrawal & Srikant 1994). Real data sets are taken from the University of California at Irvine (UCI) machine learning repository (http://archive.ics.uci.edu/ml/datasets.html) and the University of Regina (http://www2.cs.uregina.ca/~dbd/cs831/notes/itemsets/datasets.php). For synthetic data set, T10I5D100K where *T* represents the average size of the transaction is 10, *I* represents the average size of the maximal frequent item set is 5 and the number of transactions is 100,000. In this experiment, T8I5D100K, T6I4D100K, Zoo and TicTacToe datasets are considered.

Different support values are applied on these data sets to check how many nodes are tested, and the numbers of individuals are generated to get the exact number of maximal frequent item sets, run times, and so on. Here run time is the total execution time. The purpose of this new approach is for converging to a solution as fast as possible. A full experiment on these data sets is conducted, demonstrating the proposed method's ability to yield solutions rapidly by accessing the databases for fewer numbers of nodes in a lexicographic tree. Unlike Apriori, the proposed method generates an individual, X, in any level which satisfies a minimum support value, then all the other subsets of X in any level will be automatically pruned. This dramatically reduces the time for accessing a large dataset. This is also true the other way around: if the propose method generates an individual Y in any level which does not satisfy a minimum support value, then all the other supersets of Y in any level will be automatically pruned.

With Apriori algorithm, one would test all the nodes in a specific level and generate a candidate set. This candidate set generation needs a long time for finding maximal frequent item sets.

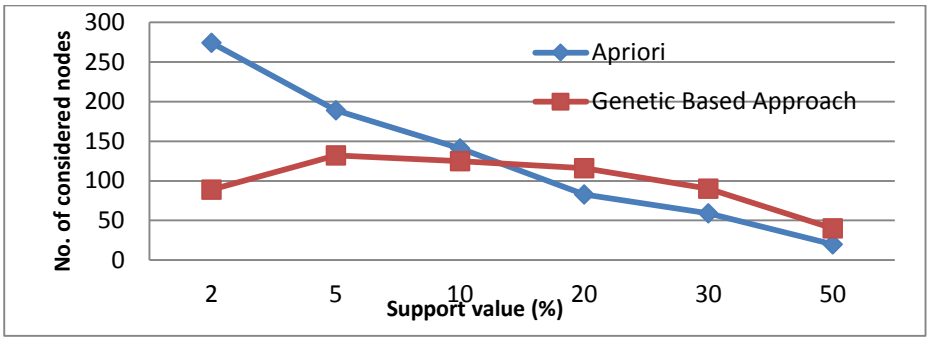


Figure 33: Zoo database

A full experiment on the Zoo dataset is shown in Figure 33. This experimental result shows the performance comparison of the proposed approach versus Apriori with different support values. The performance graph of the genetic based approach gives better or similar results than the Apriori algorithm for all support values.

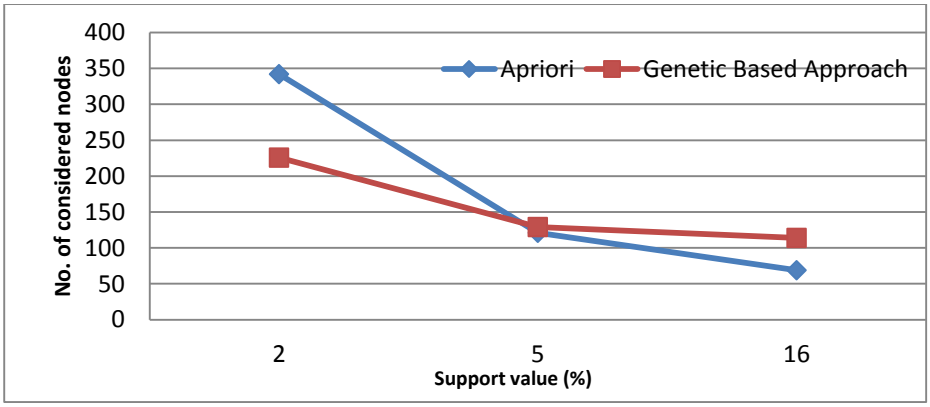


Figure 34: TicTacToe Database

A full experiment on the TicTacToe dataset is shown in Figure 34. This experimental result shows the performance comparison of the proposed approach versus Apriori with different support values. The performance graph of the genetic based approach gives better or similar results than the Apriori algorithm for all support values.

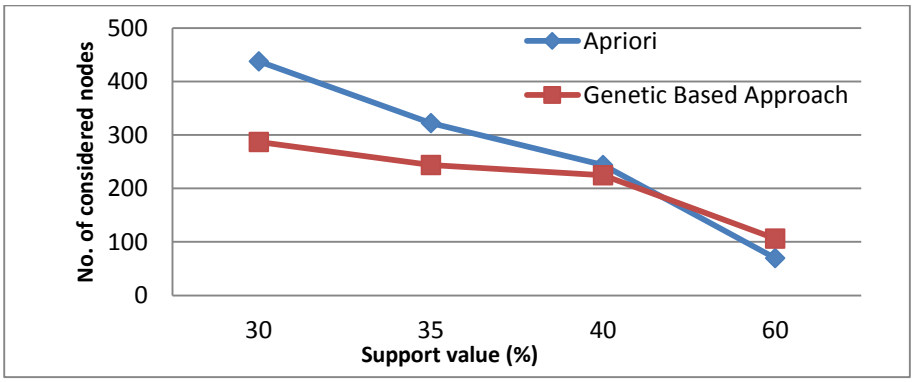


Figure 35: T8I5D100K

A full experiment on the T8I5D100K database is shown in Figure 35. This experimental result shows the performance comparison of the proposed approach versus Apriori with different support values. The performance graph of the genetic based approach gives better or similar results than the Apriori algorithm for all support values.

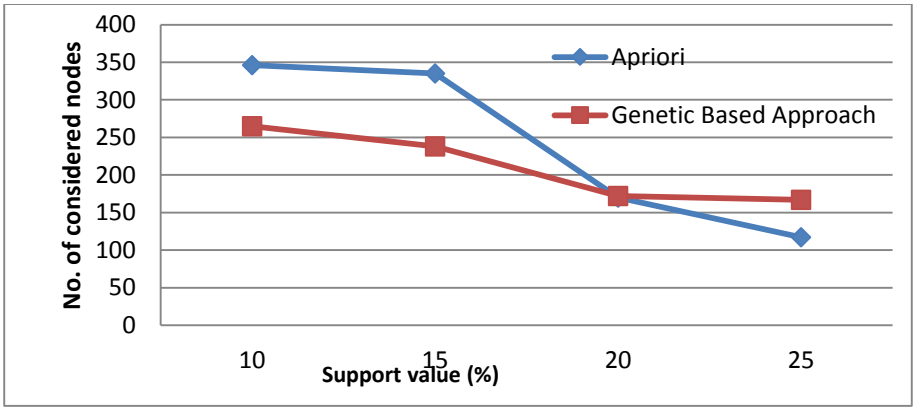


Figure 36: T6I4D100K

A full experiment on the T6I4D100K database is shown in Figure 36. This experimental result shows the performance comparison of the proposed approach versus Apriori algorithm with different support values. The performance graph of the genetic based approach gives better or similar results than the Apriori algorithm for all support values.

From the experimental results it could be concluded that the proposed mining algorithm calculates the support value for fewer nodes than the conventional Apriori algorithm, especially when the support value is low. But for higher support values, Apriori gets the solution at level k which is near to the root node in the lexicographic tree. In that case it considers fewer number of candidate item sets than the proposed algorithm. The length of the maximal frequent item sets depends on the support value. A Lower support value provides longer patterns. From longer patterns, users can get a better idea about

the relationships among frequent item sets. In that case the proposed mining algorithm outperforms the conventional Apriori algorithm.

5.2.7 Conclusion

In this thesis, a new approach (named GeneticMax) is proposed based on a genetic algorithm to mine maximal frequent item sets in an efficient way. This algorithm has been implemented and its performance has been studied. Thorough experiments have been conducted on different real data sets. The performance study shows that this algorithm mines different sizes of patterns in real data sets in an efficient way, outperforming other candidate pattern generation and evolutionary based algorithms. Several advantages have been demonstrated by the experimental analysis of GeneticMax algorithm in comparison with Apriori algorithm.

5.3 Experimental Results of Hybrid GeneticMax

5.3.1 Experimental Study

To evaluate the performance of the proposed method, several experiments have been carried out on different real world data sets. To present all the experiments, this section is organized as follows:

In subsection 5.3.2, a description of the proposed approach is presented.

In subsection 5.3.3, the data sets for this experimentation are introduced.

In subsection 5.3.4, the proposed approach is compared with the GeneticMax approach (Kabir et al. 2014).

5.3.2 Experiments

The experiments are carried out on an Intel(R) core i5-3210M CPU @2.50GHz, 4 GB RAM running on Windows 7 Enterprise. Microsoft Visual Studio 2012 is used to compile the code of the Hybrid GeneticMax. The program is written in C++ language. Four data sets including Plant Cell Signaling, Random Number #1, Synthetic and Zoo data sets are used to test the performance of the new GeneticMax approach. Different support values are applied on these data sets to check how many nodes are tested, and the number of chromosomes are generated to get the exact number of maximal frequent item sets, run times, and so on. Total execution time of the program is defined by the term run time. Three main features are embedded in the new approach:

1) it sorts out infrequent items from 1- item sets,
2) there is a superset-subset relationship in both positive and negative boundaries in a lexicographic tree for pruning invalid chromosomes, and
3) it incorporates a genetic algorithm which uses a global search mechanism.

The purpose of sorting out infrequent items from 1-item sets is that, it dramatically reduces the search space for finding the solution. Because if an item is infrequent all of its super item sets are infrequent. The aim of this new approach is to converge to a solution as fast as possible, especially if 1-item sets contain a reasonable amount of infrequent items and the solution resides in the deep level of the lexicographic tree instead of near the root. A full experiment of the new approach on the above mentioned data sets is conducted, demonstrating the ability of this method to yield solutions rapidly by accessing the data sets for a few numbers of nodes in a lexicographic tree.

As discussed in the previous sections, all the nodes in each level of a lexicographic tree are tested by Apriori algorithm and those nodes from a level which do not satisfy user defined support value are pruned. In GeneticMax, if it generates an individual X in any level which satisfies a user defined support value, then all other subsets of X in any level will be automatically pruned. This mechanism is also true the other way around: if GeneticMax generates an individual Y on any level which is infrequent i.e. which does not support a user defined support value, then all the supersets of Y in any level of a lexicographic tree will be automatically pruned. The Hybrid GeneticMax embeds all the features of the GeneticMax algorithm including local search mechanism for finding infrequent item sets from 1- item sets of a large data set.

5.3.3 Data Sets

The proposed algorithm is tested on different real data sets such as Plant Cell Signaling, Random Number #1, Synthetic, Zoo, and so on. These data sets are taken from the University of California at Irvine (UCI) machine learning repository (http://archive.ics.uci.edu/ml/datasets.html) and data sets of the University of Regina (http://www2.cs.uregina.ca/~dbd/cs831/notes/itemsets/datasets.php).

5.3.4 Comparative Analysis of Hybrid GeneticMax with GeneticMax

Several experiments are conducted on different data sets for evaluation purposes. From the experimental results as shown in Table 4, it can be concluded that the Hybrid Ge-

neticMax considers less number of nodes to get the solution than GeneticMax, which gives the same solution by considering a larger number of nodes. The above statement is true if there are a reasonable amount of infrequent items in 1- item sets. If 1- item sets do not contain any infrequent items, then the local search mechanism of Hybrid GeneticMax does work and in this case Hybrid GeneticMax performs the same as the GeneticMax algorithm.

Table 4: Number of nodes of a lexicographic tree of a Plant Cell Signaling data set, are used for getting the solution for GeneticMax and Hybrid GeneticMax algorithms

minsupp (in %)	GeneticMax	Hybrid GeneticMax
0.95	13273	76
0.8	15761	163
0.6	15761	163
0.3	15761	294
0.2	16358	2086
0.15	20151	7537
0.1	22624	8956

In this thesis, CPU time and I/O time are both included by run time. Figure 37-40 shows the run time behaviour of GeneticMax and Hybrid GeneticMax. CPU time (Run time) is needed by the existing mining approaches for calculating the support value of examined nodes. The efficiency of an algorithm depends on how many number of frequent or infrequent item sets it considers to get the final solution i.e maximal frequent item sets (Kuo & Shih 2007). For both algorithms, the same number of chromosomes are generated. However, GeneticMax accesses the data set for calculating the support value for a large number of nodes, whereas Hybrid GeneticMax considers a smaller number of nodes to get the solution. For this reason Hybrid GeneticMax takes a smaller amount of time than GeneticMax to converge to a solution.

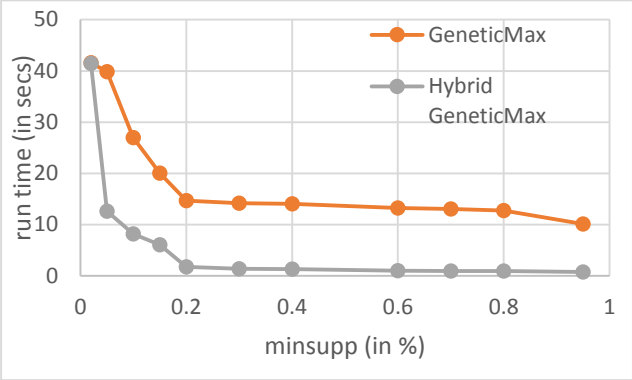


Figure 37: Performance comparison of GeneticMax versus Hybrid GeneticMax with different support values for Plant Cell Signaling data set

The performance graph of the Hybrid GeneticMax algorithm gives better results than the GeneticMax algorithm for all support values which is shown in Figure 37.

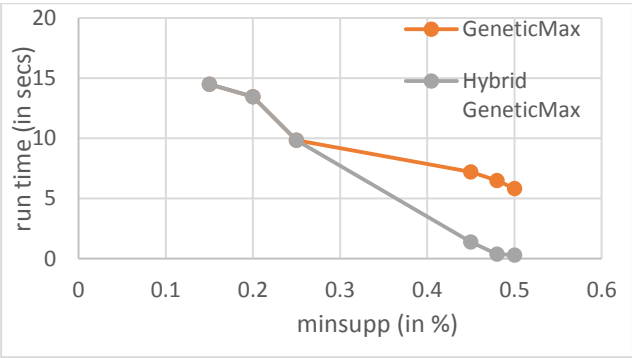


Figure 38: Performance comparison of GeneticMax versus Hybrid GeneticMax with different support values for Random Numbers #1 data set

The performance graph of the Hybrid GeneticMax algorithm gives better or similar results than the GeneticMax algorithm for all support values which is shown in Figure 38.

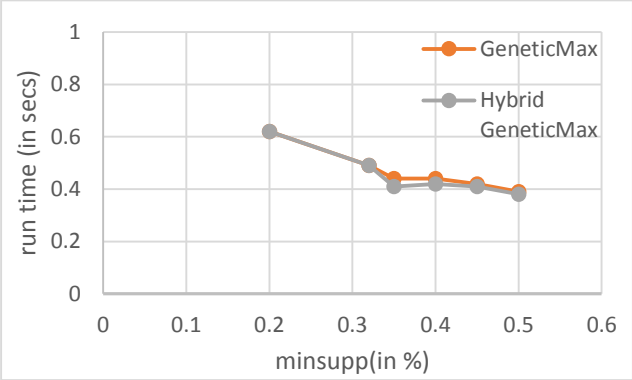


Figure 39: Performance comparison of GeneticMax versus Hybrid GeneticMax with different support values for Synthetic #3 data set

The performance graph of the Hybrid GeneticMax algorithm gives better or similar results than the GeneticMax algorithm for all support values which is shown in Figure 39.

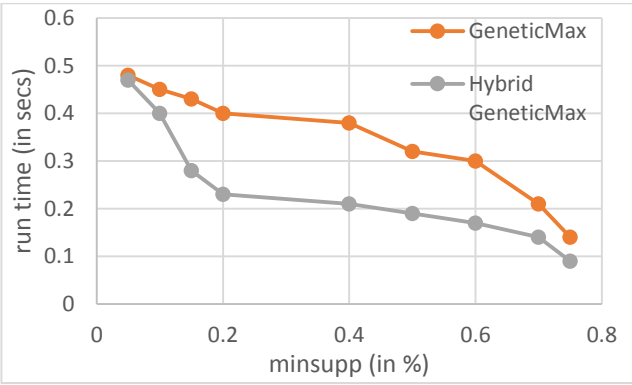


Figure 40: Performance comparison of GeneticMax versus Hybrid GeneticMax with different support values for Zoo data set

The performance graph of the Hybrid GeneticMax algorithm gives better results than the GeneticMax algorithm for all support values which is shown in Figure 40.

From the performance graph it can be concluded that Hybrid GeneticMax outperforms the GeneticMax algorithm especially if 1- item sets contain a reasonable amount of infrequent item sets. Both algorithms are tested on various data sets for different support values and the experimental results show that the Hybrid GeneticMax performs better than the GeneticMax algorithm until it reaches a certain threshold value of minsupp. After that 1 item sets do not contain any infrequent items, so the Hybrid GeneticMax performs the same as the GeneticMax i.e. both algorithms access the same number of nodes to get the solution and the computational time is the same as well.

5.3.5 Conclusion

The Hybrid GeneticMax algorithm used the technique of a local search and a genetic algorithm to mine maximal frequent item sets in an efficient way. Thorough experiments have been carried out on different real data sets for evaluating the performance of the GeneticMax and Hybrid GeneticMax algorithm. The experimental results demonstrate several advantages of the proposed algorithm, in comparison with the GeneticMax algorithm. The experimental analysis of the Hybrid GeneticMax shows the effect of a local search along with a global search mechanism, and it compared the results with the GeneticMax algorithm. From the experimental results, it can be concluded that Hybrid GeneticMax outperforms the GeneticMax algorithm, if there are a reasonable amount of infrequent items in 1-item sets. For a certain threshold value, if there are no infrequent items in 1-item sets, then this approach performs similar to the GeneticMax algorithm.

5.4 Experimental Results of PSO

5.4.1 Experiments

The experiments are performed on an Intel(R) core i5-3210M CPU @2.50GHz, 4 GB RAM running on Windows 7 Enterprise. The algorithm is written in C++ language. Microsoft Visual Studio 2012 is used to compile the code.

5.4.2 Evaluation of the Experiments

Initially the proposed algorithm is tested on a small database that contains 5 items for a large number of transactions. A few parameters are considered for this experiment as shown in the following table.

Table 5: Parameters for association rule mining algorithm using PSO

Specification	Value
Number of particles	2
User define support value	40%
User define confidence value	50%
Selection of the path	Randomly select the next path by following the movement message of gbest and pbest

For selecting the next path of a particle, particle depends on the movement message of "gbest" and the direction of current "pbest" value. For this experiment after getting "gbest" and "pbest" value, a particle changes its position randomly and updates its "pbest" accordingly. This random position change will help avoid the "local optima" problem.

Table 6: Frequent item sets with support and confidence value

Database	Records	Items	Support (%)	Item	Frequent Item sets	Confidence	Remarks
1000×5	1000	5	40	1	{1,2,3,4}, {1,3,4,5}	{40%}, {40%}	
				2	{2,3,4}	{70%}	
				3	{3,4,5}	{40%}	
				4	{4,5}	{50%}	
				5	{5}		Single item, does not generate rules

Table 6 shows the frequent item sets that are generated under a user defined support value. From this result it can be concluded that, item sets {1,2,3,4}, {1,3,4,5} contain maximum items. The item column in Table 6 refers to the item for which the position is fixed. For item 1, the particle's search space considers all the superset which will be generated from 1. For item 3, the search space contain {3,4}, {3,5} and {3,4,5} positions. From the above result it can be seen that for item 1, the generated frequent item sets are {1,2,3,4},{1,3,4,5}, whereas the generated frequent item set is {2,3,4} for item 2, which is also the subset of the generated frequent item sets of item 1. This approach considers the pruning strategies that, all the subsets of a frequent item set will be pruned. If the whole search space is not subdivided by the item number then some interesting rules could be missed. For example, the confidence value of item set {1,2,3,4} is 40%. On the other hand, the confidence value of item set {2,3,4} which is the subset of item set {1,2,3,4} is 70%. That is, item set {2,3,4} can generate a strong rule. For this reason

the whole search space is subdivided by the item number. Otherwise it could miss some interesting rules.

Table 7: Generated Strong Rules

Database	Frequent Item sets	Confidence	Remarks
1000×5	{1,2,3,4}, {1,3,4,5}	{40%},{40%}	No rules generated
	{2,3,4}	{70%}	2→3,4 2,3→4
	{3,4,5}	{40%}	No rules generated
	{4,5}	{50%}	4→5
	{5}		Single item, does not generate rules

Table 8: Infrequent item sets

Database	Support (%)	Item	Infrequent Item sets
1000×5	<40%	1	{1,2,3,4,5}
		2	{2,3,4,5}
		3	None
		4	None
		5	None

Table 7 shows the generated strong rules which satisfy the user defined support and confidence values. This mining approach for each item number generates frequent and infrequent item sets. Table 8 shows the infrequent item sets. Users can generate association rules from infrequent item sets.

5.4.3 Conclusion

To mine association rules for both frequent and infrequent item sets in an efficient way, in this thesis a PSO based approach is proposed. The experimental results demonstrates several advantages of the proposed method. From the experimental result, it can be concluded that this approach shows the power of using a heuristic algorithm for generating

association rules for frequent item sets along with infrequent ones from a lexicographic tree.

5.5 Experimental Results of ARMGAAM

5.5.1 Experimental Study

Several experiments are carried out on different data sets for evaluating the performance of the proposed method. To present the experiments, this section is organized as follows:

In subsection 5.5.2, the data sets which are used for this experiments are introduced.

In subsection 5.5.3, the specifications of the data sets and the parameters that are considered for different methods are presented.

In subsection 5.5.4, the performance of the proposed approach with another GA based approach for mining BARs named, ARMGA(Yan et al. 2009; Yan et al. 2005) is compared.

In subsection 5.5.5, the obtained results of the proposed method with two other classical algorithms Apriori (Agrawal & Srikant 1994; Borgelt 2003) and Eclat (Hipp et al. 2000; Zaki 2000) are compared.

In subsection 5.5.6, some of the rules that are obtained by the proposed method are analysed.

In subsection 5.5.7, the scalability of the proposed method is studied.

5.5.2 Data Sets

In order to evaluate the performance of the proposed method, this experimental study is carried out on four real world data sets, with the number of items and records ranging from 21 to 73 and 277 to 5456, respectively. These data sets are taken from the University of California at Irvine (UCI) machine learning repository (http://archive.ics.uci.edu/ml/datasets.html).

5.5.3 Experiments

The specifications of data sets are summarized in Table 9, the number of attributes is represented by Attributes and the number of examples in a data set is represented by Examples. In these experiments, the performance of the proposed approach is compared

Table 9: Datasets that are considered for the experimental analysis

	Chess (Kr vs Kp)	Breast Cancer	Car Eval.	Plant Cell Sig.
Attributes	73	51	21	43
Examples	3196	277	1728	5456

with the three other algorithms, named ARMGA (Yan et al. 2009; Yan et al. 2005), Apriori (Agrawal & Srikant 1994; Borgelt 2003), and Eclat (Hipp et al. 2000; Zaki 2000).

Table 10: Parameters considered for different algorithms

ARMGA	popsize = 100, P_{sp} = 0.95,P_{cp} = 0.85,P_{mp} = 0.01,α = 0.01, maxloop = 5~25, k =3
Apriori	min_supp = 0.01, min_conf = 0.1
Eclat	min_supp = 0.01, min_conf = 0.1
ARMGAAM	popsize = 100, P_{sp} = 0.95, P_{cp} = 0.85, P_{mp} = variable, α = 5%, maxloop = 5~25, k = 3

Table 10 shows the parameters of the analyzed algorithms. In this experiment, instead of searching specific values standard common parameters are used by the proposed approach which work well for facilitating the comparisons. The parameters of the remaining algorithms are selected based on the recommendations of the corresponding authors of each approach. For all the experiments conducted in this study, the results shown in the table for the GA based approaches always refer to those non-dominated rules which consider positive dependence among the item sets with potential interest. For developing the different experiments, the average results of three runs for each data set are considered.

5.5.4 Comparative Analysis of the Proposed Method with Other Evolutionary Algorithm Based Approach

The performance of the proposed approach against an evolutionary algorithm for mining BARs, the ARMGA (Yan et al. 2009; Yan et al. 2005) algorithm, is shown in Table 11, where #R represents the number of generated BARs, Av_{supp}, Av_{conf}, Av_{lift}, $Av_{netconf}$, Av_{CP} are average support, confidence, lift, netconf, conditional probability, respectively. As with ARMGAAM, it generates a reduced set of BARs than ARMGA for all the data sets. From an analysis of the results shown in Table 11, it can be concluded that for all data sets the rules obtained by the proposed approach show improvements in almost all the interestingness measures over those obtained rules generated by ARMGA.

Table 11: Results obtained for all the data sets in comparison with ARMGA

Algorithms	#R	Av_{sup}	Av_{conf}	Av_{lift}	$Av_{netconf}$	Av_{CP}
Chess(King-Rook vs King-Pawn)						
ARMGA	**47**	0.09	0.85	2.71	0.34	0.72
ARMGAAM	32	**0.10**	**0.86**	**3.39**	**0.41**	**0.78**
Car Evaluation						
ARMGA	**38**	**0.02**	0.08	**1**	**0.0001**	**0.0001**
ARMGAAM	26	**0.02**	**0.1**	**1**	**0.0001**	**0.0001**
Plant Cell Signaling						
ARMGA	**18**	**0.63**	**0.86**	1.81	0.54	0.63
ARMGAAM	15	0.6	0.84	**1.97**	**0.56**	**0.65**
Breast Cancer						
ARMGA	**9**	**0.03**	0.17	3.19	0.1	0.1
ARMGAAM	5	0.02	**0.29**	**8.36**	**0.27**	**0.26**

5.5.5 Comparative Analysis of the Proposed Method with Classical Algorithms

The comparison of the proposed method with the other two classical rule mining algorithms, Apriori (Agrawal & Srikant 1994; Borgelt 2003) and Eclat (Hipp et al. 2000; Zaki 2000), is shown in Table 12. In most data sets, Apriori and Eclat generate a large set of BARs, have high support and confidence values, but a low value for each of the

Table 12: Results obtained for all the data sets in comparison with the classical algorithms

Algorithms	#R	Av_{sup}	Av_{conf}	Av_{lift}	$Av_{netconf}$	Av_{CP}
Chess(King-Rook vs King-Pawn)						
Apriori	**17592**	**0.68**	**0.88**	1.01	0.04	0.05
Eclat	**17592**	**0.68**	**0.88**	1.01	0.04	0.05
ARMGAAM	32	0.10	0.86	**3.39**	**0.41**	**0.78**
Car Evaluation						
Apriori	**5082**	**0.02**	**0.18**	**1**	-4.17E-09	-5.05E-09
Eclat	**5082**	**0.02**	**0.18**	**1**	-4.17E-09	-5.05E-09
ARMGAAM	26	**0.02**	0.1	**1**	**0.0001**	**0.0001**
Plant Cell Signaling						
Apriori	**12147**	**0.9**	**0.96**	1.02	∞	∞

Eclat	**12147**	**0.9**	**0.96**	1.02	∞	∞
ARMGAAM	15	0.6	0.84	**1.97**	0.56	0.65
Breast Cancer						
Apriori	**94**	**0.29**	**0.85**	1.14	0.16	**0.4**
Eclat	**94**	**0.29**	**0.85**	1.14	0.16	**0.4**
ARMGAAM	5	0.02	0.29	**8.36**	**0.27**	0.26

remaining measures due to the fact those classical rule mining algorithms generate a huge number of misleading rules. For the Plant Cell Signaling data set, the value ∞ shown in the table represents the maximum value in some measures. This value is generated due to the presence of a large number of trivial rules. By contrast, the proposed approach allows users to obtain a reduced set of BARs which have similar or low values for support and confidence measures but high or similar values for the rest of the measures.

5.5.6 Rules Obtained by the Proposed Method

Some useful and interesting rules which are generated by the proposed approach are shown in Table 13. Two of the generated rules from Table 12 are interpreted in Table 13.

Table 13: Some of the obtained Rules of a car evaluation data set

Dataset	Rules
Car Evaluation	R1: The buying price of a car is high only if it can carry 2 persons and the size of the luggage boot is big. R2: The car of 5 or more doors has a medium safety only if it carries more persons.

5.5.7 Scalability Analysis

Several experiments are carried out for analysing the scalability of the proposed approach for the Chess (King-Rook vs King-Pawn) data set. The experiments are performed on an Intel(R) core i5-3210M CPU @2.50GHz, 4 GB RAM running on Windows 7 Enterprise. The average runtime expended by the algorithms, when the number

Table 14: Expended runtime (in seconds) of all the algorithms when the number of attributes is increased within a data set Chess (King-Rook vs King-Pawn)

Number of attributes					
Algorithms	15	25	35	55	73
ARMGA	5.1	6.3	7.1	6.97	6.8
Apriori	0.6	1.01	2.06	4.6	56.74
Eclat	0.6	1.05	2.76	4.9	59.16
ARMGAAM	5.87	4.36	4.92	6.81	5.78

Table 15: Expended runtime (in seconds) of all the algorithms when the number of examples is increased within a data set Chess (King-Rook vs King-Pawn)

Number of examples					
Algorithms	20%	40%	60%	80%	100%
ARMGA	2.61	3.1	3.8	4.1	6.8
Apriori	18.61	28.68	44.79	48.12	56.74
Eclat	22.16	30.56	45.78	44.19	59.16
ARMGAAM	2.85	2.81	4.1	4.43	5.78

of attributes and examples are increased is shown in Tables 14 and 15, respectively. From Figure 41 it can be seen that all the evolutionary algorithms scale quite linearly whereas the classical algorithms, Apriori and Eclat, increase exponentially especially when the number of attributes is increased.

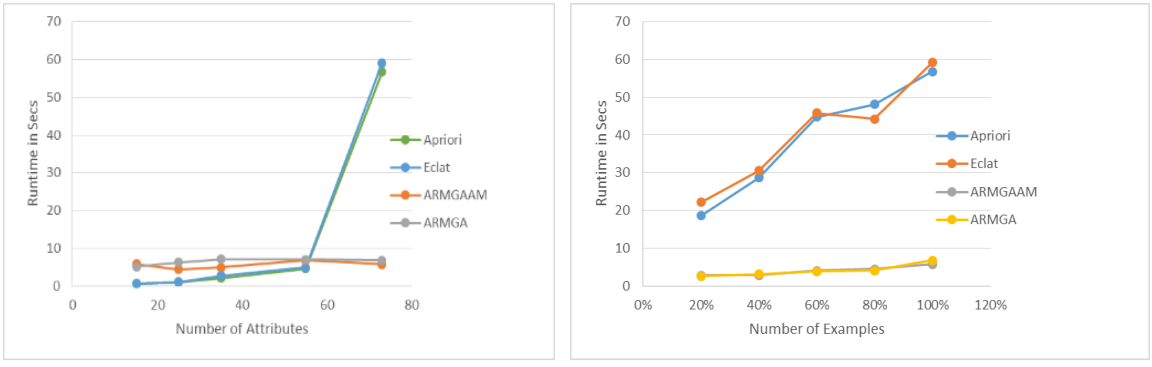


Figure 41: Required runtime for different algorithms for different number of attributes and examples in a Chess (King-Rook vs King-Pawn) data set

5.5.8 Conclusion

ARMGAAM is used to mine a reduced set of Boolean association rules. The generated BARs are interesting, easy to understand and maximizing three objectives: lift, net confidence and conditional probability. To accomplish this, the proposed algorithm extends the ARMGA algorithm for performing evolutionary learning and selection of a condition of each rule. From the experimental results obtained over four real world data sets, it can be concluded that the proposed approach allows users to mine a reduced set of BARs with good trade- off among the number of generated rules, support, confidence, lift, net confidence and conditional probability of all the data sets. Moreover, the obtained rules are strong, showing a strong relationship among the item sets and solving the problem of the support-confidence framework. Finally, the proposed approach has a good computational cost and scalability when the problem size increases.

5.6 Experimental Results of MBAREA

5.6.1 Experimental Study

Several experiments are carried out on different data sets for evaluating the performance of the proposed method. In this section the following studies are performed:

In subsection 5.6.2, the data sets which are considered for this analysis are introduced.

In subsection 5.6.3, the specifications of the data sets and the parameters which are used for running the methods are presented.

In subsection 5.6.4, the performance of the proposed method is compared with two other evolutionary approaches, ARMGA(Yan et al. 2009; Yan et al. 2005) and ARMMGA (Qodmanan et al. 2011).

In subsection 5.6.5, the scalability of the proposed approach is explained.

In subsection 5.6.6, some of the obtained rules by the proposed method are analysed.

5.6.2 Data Sets

In order to assess the performance of the proposed method, an experimental analysis using six real world data sets is presented. The number of attributes of the data sets ranges from 23 to 118 and the number of records from 267 to 12,960. These data sets

are available in the UCI machine learning repository (http://archive.ics.uci.edu/ml/datasets.html).

5.6.3 Experiments

Several experiments are carried out on the different data sets for analysing the efficiency of the proposed algorithm. For testing the proposed algorithm and comparing the result with ARMGA and ARMMGA approaches, six real world data sets are considered.

Table 16: Data sets considered for the experimental analysis

	Mushroom	Balance Scale	Nursery	Monk's Problems	Solar Flare	SPECT Heart
Attributes (B)	118	23	32	19	50	46
Records	8124	625	12960	431	1066	267

Table 16 summarizes the specifications of those data sets, where Attributes (B) represents the number of Boolean attributes and Records, the number of records.

Table 17: Parameters considered for running the algorithms

Algorithms	**Parameters**
ARMMGA	Popsize= 100, P_{sel} = 0.95, P_{cro} = 0.85, P_{mut} = 0.01, db = 0.01, k =3.
ARMGA	Popsize= 100, P_{sel} = 0.95, P_{cro} = 0.85, P_{mut} = 0.01, α = 0.01, k =3.
MBAREA	Popsize= 100, P_{sel} = 0.95, P_{cro} = 0.85, P_{mut} = [100-{(100/δ)*n}] %, δ = 5, n= 1~ δ, k =3, α = 0.01.

The parameters, which are used for running the algorithms are shown in Table 17. For ARMMGA and ARMGA, the parameters are selected according to the recommendations of each proposal.

5.6.4 Comparative Analysis of the Proposed Method with Other Evolutionary Algorithm Based Approaches

As described in section 4.6.3, a class based mutation method is applied for the proposed approach and the probability of mutation ratio is decreased with respect to the class of chromosomes in a population.

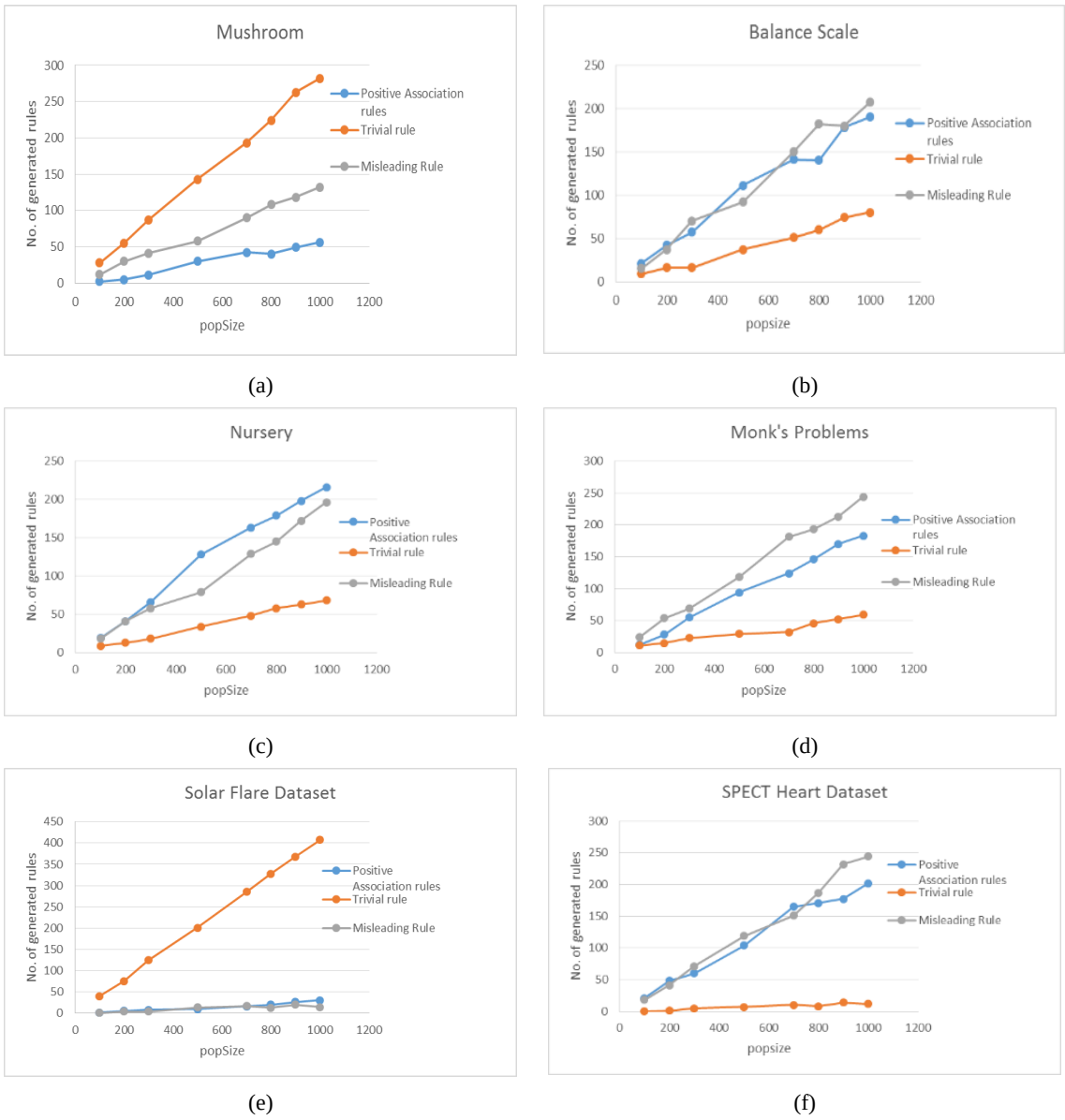


Figure 42: Different types of rules for different data sets are generated by ARMMGA because of using weak constraint

Because of using a weak constraint function (Qodmanan et al. 2011), ARMMGA generates positive association rules including misleading and trivial rules, which are shown in Figure 42. In this experiment single evaluation result is considered for different popsizes.

Table 18: Results obtained by evolutionary algorithms for different data sets

Algorithms	**#Rules**	**Av_{supp}**	**Av_{conf}**	**Av_{lift}**	**Av_{int}**	**CP**
SPECT Heart						
ARMMGA	9	**0.29**	**0.87**	1.25	**0.0005**	0.2
ARMGA	**37**	0.25	0.6	1.5	0.0004	0.3

MBAREA	11	0.1	0.83	**1.88**	**0.0005**	**0.68**
Monk's Problems						
ARMMGA	7	0.04	0.64	1.88	8.84E-06	0.19
ARMGA	**23**	**0.06**	0.4	3.06	4.27E-05	0.24
MBAREA	15	**0.06**	**0.8**	**6.46**	**0.0001**	**0.76**
Balance Scale						
ARMMGA	5	0.02	0.72	1.56	2.24E-06	0.35
ARMGA	**15**	**0.03**	0.51	**2.82**	**6.02E-06**	0.38
MBAREA	8	**0.03**	**0.77**	1.678	2.60E-06	**0.58**
Solar Flare						
ARMMGA	7	**0.005**	**1**	236.88	5.86E-07	0.5
ARMGA	**12**	0.003	0.59	113.19	2.16E-06	0.6
MBAREA	10	**0.005**	0.947	**237.31**	**4.10E-06**	**0.94**
Mushroom						
ARMMGA	10	**0.028**	0.38	12.09	3.75E-07	0.40
ARMGA	**29**	0.018	0.47	6.26	2.29E-07	0.37
MBAREA	18	0.002	**0.97**	**12.74**	**1.47E-06**	**0.94**
Nursery						
ARMMGA	4	**0.028**	0.52	1.19	1.15E-07	0.2
ARMGA	**14**	0.02	0.5	3.54	**3.28E07**	0.37
MBAREA	6	0.01	**1**	**8**	1.47E-11	**1**

The performance of the proposed approach against other algorithms is shown in Table 18, where #Rules is the number of generated rules and Av_{supp}, AV_{conf}, AV_{lift}, Av_{int} and CP are average support, confidence, lift, interest and conditional probability, respectively. In order to develop the different experimental analysis, the average results of three runs for each data set are considered.

The rules obtained by the proposed approach presents better or similar values for different measures than the rules obtained by other algorithms. As with ARMMGA, it generates a smaller number of rules but some of those are misleading or trivial rules. For some data sets, ARMGA obtains good average support but low values for the rest of the measures. For all data sets, the values of average confidence, lift, interest and condition-

al probability of the rules generated by MBAREA are better than or similar to the rules generated by other algorithms. Moreover, the rules generated by the proposed approach are not misleading or trivial.

5.6.5 Scalability Analysis

To analyse the scalability of the proposed algorithm, several experiments are carried out on the Nursery data set. The experiments are performed on an Intel(R) core i5-3210M CPU @2.50GHz, 4 GB RAM running on Windows 7 Enterprise.

Table 19: Runtime (in secs) needed for different attributes of the Nursery data set

Algorithms	**Number of attributes**				
	8	12	20	25	32
ARMMGA	10.68	8.34	11.4	10.87	13.13
ARMGA	9.15	10.88	12.25	13.37	13.85
MBAREA	9.1	9.43	10.09	14.35	10.88

The average runtime required by the algorithms, as the number of attributes and examples are increased is shown in Tables 19 and 20, respectively. From the experimental result it can be concluded that all the algorithms scale quite linearly, however in most of the cases MBAREA takes less time than other algorithms.

Table 20: Runtime (in secs) needed for increasing number of examples of the Nursery data set

Algorithms	**Number of examples**				
	20%	40%	60%	80%	100%
ARMMGA	8	9.45	10	10.4	13.13
ARMGA	3.77	9.17	10.46	13.24	13.85
MBAREA	3.09	8.32	8.58	12.84	10.88

5.6.6 Rules Obtained by the Proposed Method

Some useful and interesting BARs which are generated by MBAREA are shown in Table 21. These are the rules with positive dependence among the item sets and that have maximum value of other objectives such as lift, CP and so on.

Table 21: Rules obtained by the proposed method for different data sets

Data Sets	Rules	Confidence	Lift	CP
SPECT Heart	37,26 → 16	1	1.74	1
Mushroom	98,86 → 34	1	1.19	1
Nursery	28 → 19,12	1	8	1

For example, the rule 28 → 19, 12 of the Nursery data set in Table 21 could be interpreted as follows: the decision will be to "recommend" the application only if the financial condition of a parent is maximum value of other objectives such as lift, CP and so on.

5.6.7 Conclusion

The proposed approach, MBAREA, is a new evolutionary algorithm for mining a reduced set of positive BARs. The generated rules are interesting, easy to understand and maximize two objectives, performance and interestingness. To accomplish this, MBAREA extends the existing ARMGA and ARMMGA algorithms for evolutionary learning and selection of a condition of each rule. MBAREA introduces a class based mutation method to the evolutionary model and a best population technique to improve the diversity of the generated rules and to store all the non-dominated rules which are generated in the intermediate generation of a population. Analyzing the results obtained over six real world data sets, it can be concluded that the generated rules maintain a good trade-off between the number of rules, confidence, conditional probability, interest and lift values in all the data sets. Finally, the experimental results show that MBAREA has a good computational cost and scales well when the problem size is increased.

5.7 Experimental Results of MSGA

5.7.1 Experimental Study

In this section several experiments are conducted on different real world data sets for evaluating the effectiveness of the MSGA method. For presenting all the experiments, this section is organized as follows:

In subsection 5.7.2, the data sets which are used for this experiments are introduced.

In subsection 5.7.3, the specifications of the data sets and the configuration parameters which are used for this analysis are presented.

In subsection 5.7.4, the performance of the proposed method is compared with the difference single seeds based approaches.

In subsection 5.7.5, the convergence of the proposed algorithm and different single seeds based methods for various crossover and mutation operators is analyzed.

5.7.2 Data Sets

To analyse the performance of the proposed method, several experiments are carried out on different real world data sets. For evaluating the proposed approach and comparing the results with a simple single seed based genetic algorithm, four real world data sets are considered, which are taken from the UCI machine learning repository (http://archive.ics.uci.edu/ml/datasets.html).

5.7.3 Experiments

As mentioned in section 4.7.6, the categorical attributes of a data set are mapped into Boolean attributes. The specifications of these mapped data sets are summarized in

Table 22: The specifications of data sets

	Breast Cancer	**Solar Flare**	**Monk's Problems**	**Mushroom**
Attributes	51	50	19	118
Records	277	1066	431	8124

Table 22, where the number of Boolean attributes and records of a data set are represented by "Attributes" and "Records", respectively.

Table 23: The parameters used for running the MSGA

	Breast cancer	**Solar Flare**	**Monk's Problems**	**Mushroom**
Population Size (popsize)	100	100	100	100
Selection Probability (sp)	0.95	0.95	0.95	0.95
Crossover Probability (cp)	0.85	0.85	0.85	0.85
Mutation Probability (mp)	0.01	0.01	0.01	0.01
Rule Length (k)	3	3	3	3
No. of Generations	55	55	55	55
No. of Archives	4	4	4	4

Table 24: The parameters used for running a single seed based SGA

	Breast cancer	**Solar Flare**	**Monk's Problems**	**Mushroom**
Population Size (popsize)	100	100	100	100
Selection Probability (sp)	0.95	0.95	0.95	0.95
Crossover Probability (cp)	0.85	0.85	0.85	0.85
Mutation Probability (mp)	0.01	0.01	0.01	0.01
Rule Length (k)	3	3	3	3
No. of Generations	55	55	55	55

Table 23 and Table 24, show the parameters which are used for running the MSGA and a single seed based simple genetic algorithm. As described in Chapter 4 (see section 4.7), multiple seeds are generated from archives and these seeds are applied to generate an initial population for mining high quality association rules, each having a maximum conditional probability of between 80%-100%. Traditionally, a simple genetic algorithm randomly generates a single seed for initializing a population.

In order to analyse the performance of MSGA over a single seed based simple genetic algorithm, seeds which are generated by MSGA for initializing a population are used separately in this experiment. The effects of different mutation and crossover operators on the initial population for evolutionary learning for generating high quality association rules are examined in four different real world data sets. In order to develop the different experiments, the average results of five runs are considered for each data set.

5.7.4 Performance Analysis of the Proposed Method with Different Single Seeds Based Methods

Since an initial population has a significant effect on generating a best population in further generation, single seeds based genetic algorithms generate a smaller number of positive association rules, which are shown in Figures 43-54. Form the experimental results it can be seen that some seeds generate a large number of rules for some mutation and crossover operators, but for other crossover operators these seeds generate a smaller number of rules for the same mutation operators. On the other hand, the results obtained by the MSGA present better or similarly high quality rules for different muta-

tion and crossover operators for all data sets than the rules obtained by single seeds based genetic algorithms.

For the Breast Cancer data set, seed 1 and seed 4 have fitness values of 1 and 0.21, respectively. According to Figure 43, the seed 1 based genetic algorithm generates a large number of high quality rules using insertion (INS) mutation and uniform crossover operators with respect to other single seeds based genetic algorithms. Whereas, seed 4 based genetic algorithm performs better than other seeds based genetic algorithms using displacement (DISP), inversion (INV), scramble (SCM) mutation and uniform crossover operators. From the above analysis it can be concluded that, a genetic algorithm based on a single seed having a high fitness value cannot guarantee that it will generate a large number of high quality rules using different mutation and crossover operators for all data sets. This is also true for a lower fitness value based seed chromosome. On the other hand, MSGA which comprised all seeds to generate an initial population has a significant effect for further generation of best population and this approach present better or similar high quality rules using different mutation and crossover operators for all data sets, which are shown in Figures 43-54.

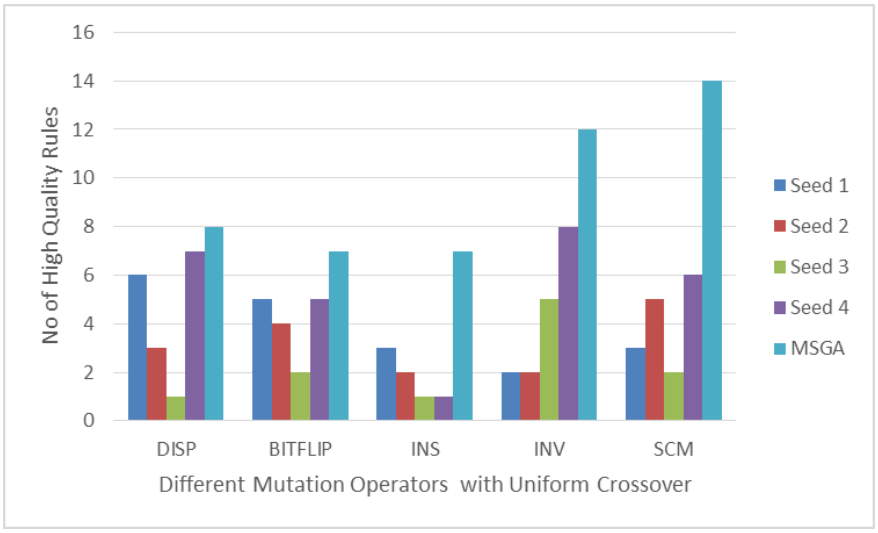


Figure 43: Performance of different seeds and MSGA for different mutation operators with uniform crossover for a Breast Cancer data set.

The performance of different seeds and MSGA for different mutation operators with uniform crossover for a Breast Cancer data set is shown in Figure 43. According to Figure 43, MSGA performs better than other single seed based genetic algorithms for all mutation operators.

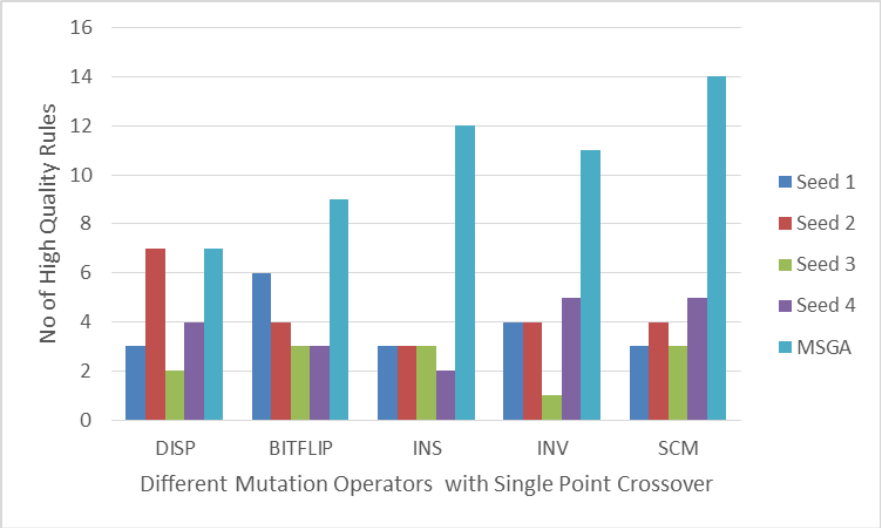


Figure 44: Performance of different seeds and MSGA for different mutation operators with single point crossover for a Breast Cancer data set.

The performance of different seeds and MSGA for different mutation operators with single point crossover for a Breast Cancer data set is shown in Figure 44. According to Figure 44, MSGA performs better than or similarly to other single seed based genetic algorithms for all mutation operators.

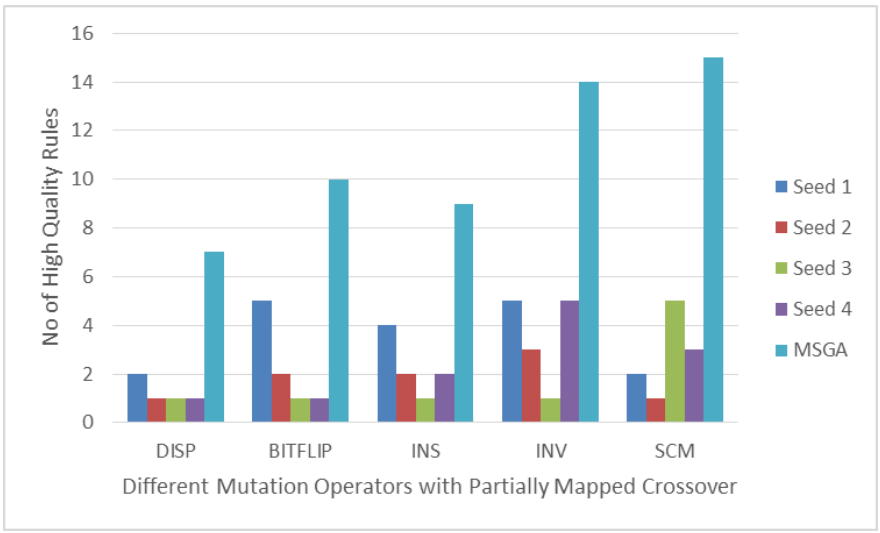


Figure 45: Performance of different seeds and MSGA for different mutation operators with partially mapped crossover for a Breast Cancer data set.

The performance of different seeds and MSGA for different mutation operators with partially mapped crossover for a Breast Cancer data set is shown in Figure 45. According to Figure 45, for all mutation operators the number of rules generated by MSGA is higher than other single seed based genetic algorithms.

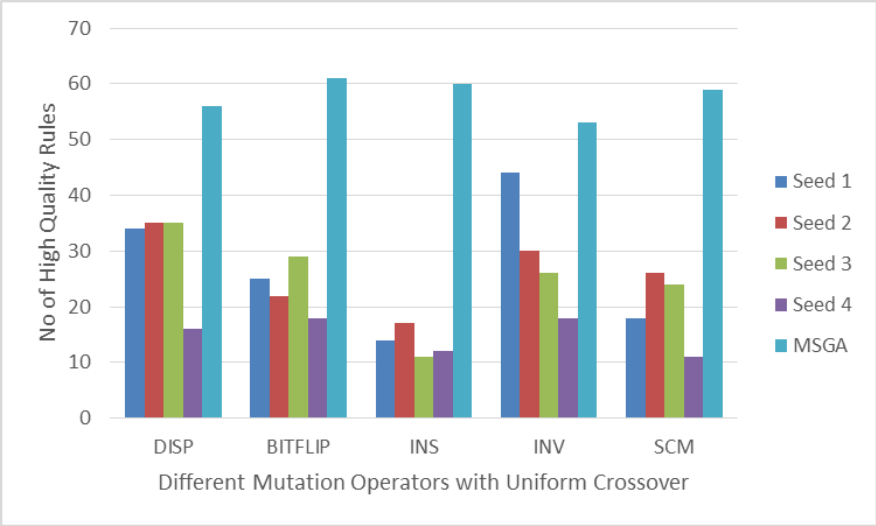


Figure 46: Performance of different seeds and MSGA for different mutation operators with uniform crossover for a Solar Flare data set.

The performance of different seeds and MSGA for different mutation operators with uniform crossover for a Solar Flare data set is shown in Figure 46. According to Figure 46, for all mutation operators the number of generated rules by MSGA is higher than other single seed based genetic algorithms.

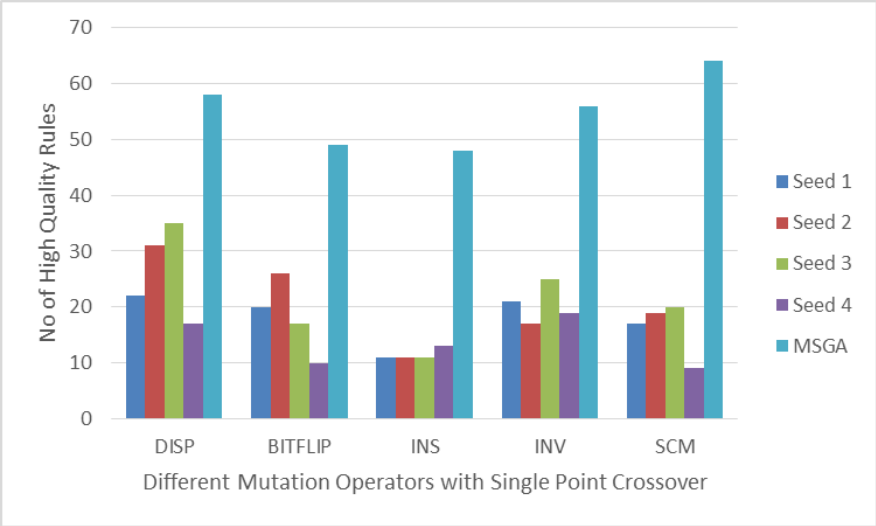


Figure 47: Performance of different seeds and MSGA for different mutation operators with single point crossover for a Solar Flare data set.

The performance of different seeds and MSGA for different mutation operators with single point crossover for a Solar Flare data set is shown in Figure 47. According to Figure 47, for all mutation operators the number of generated rules by MSGA is higher than other single seed based genetic algorithms.

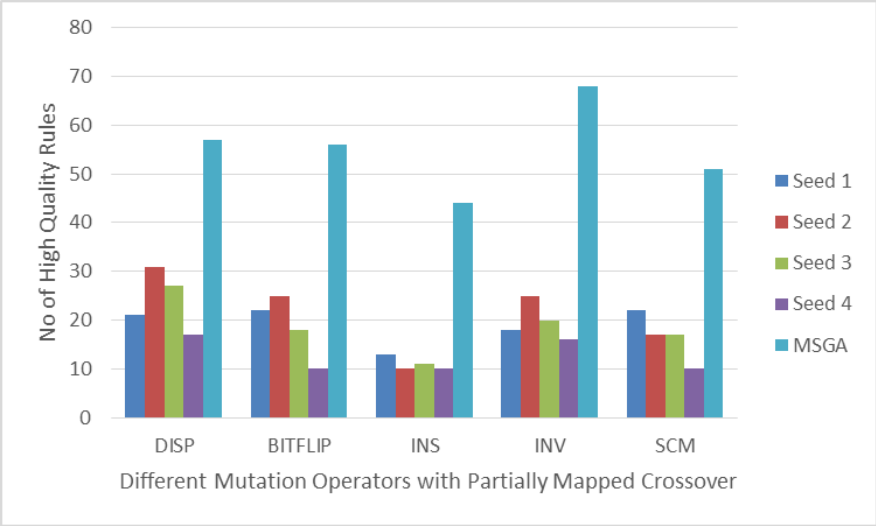


Figure 48: Performance of different seeds and MSGA for different mutation operators with partially mapped crossover for a Solar Flare data set.

The performance of different seeds and MSGA for different mutation operators with partially mapped crossover for a Solar Flare data set is shown in Figure 48. According to Figure 48, the number of generated rules by MSGA is higher than other single seed based genetic algorithms for all mutation operators.

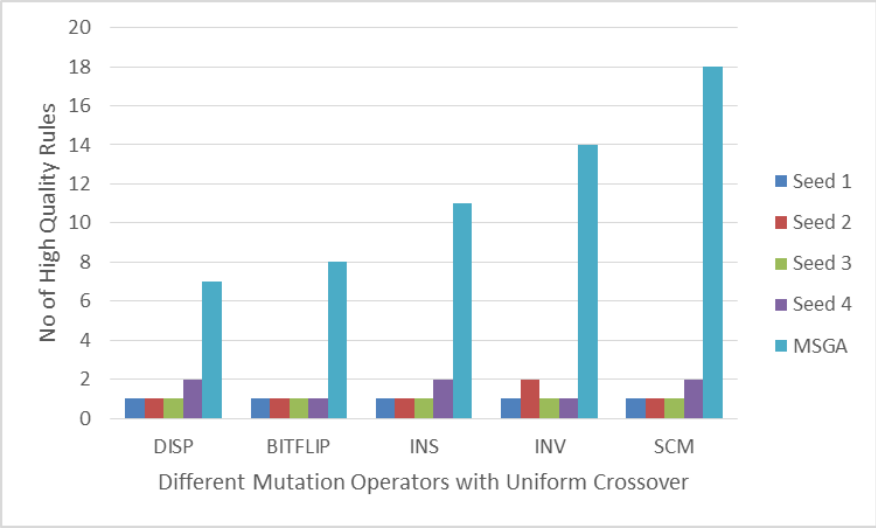


Figure 49: Performance of different seeds and MSGA for different mutation operators with uniform crossover for a Monk's Problems data set.

The performance of different seeds and MSGA for different mutation operators with uniform crossover for a Monk's Problems data set is shown in Figure 49. According to Figure 49, for all mutation operators the number of rules generated by MSGA is higher than other single seed based genetic algorithms.

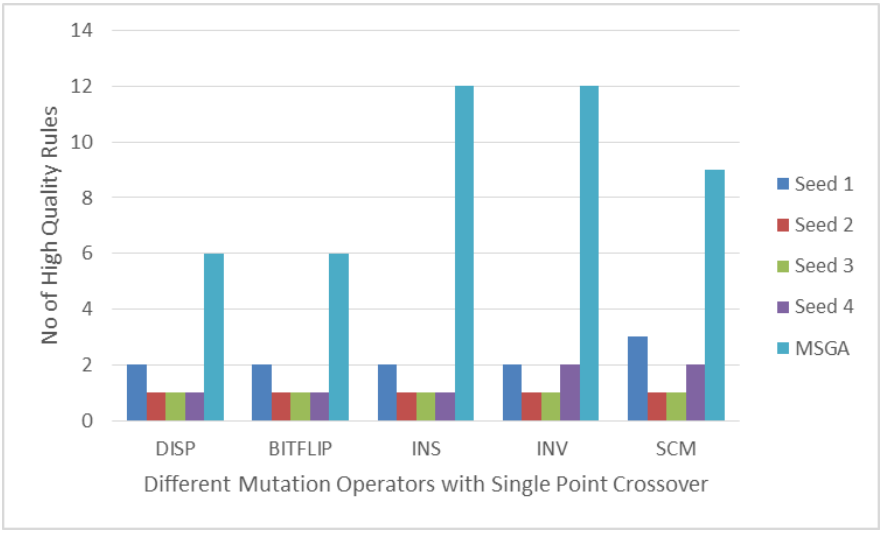


Figure 50: Performance of different seeds and MSGA for different mutation operators with single point crossover for a Monk's Problems data set.

The performance of different seeds and MSGA for different mutation operators with single point crossover for a Monk's Problems data set is shown in Figure 50. According to Figure 50, the number of generated rules by MSGA is higher than other single seed based genetic algorithms for all mutation operators.

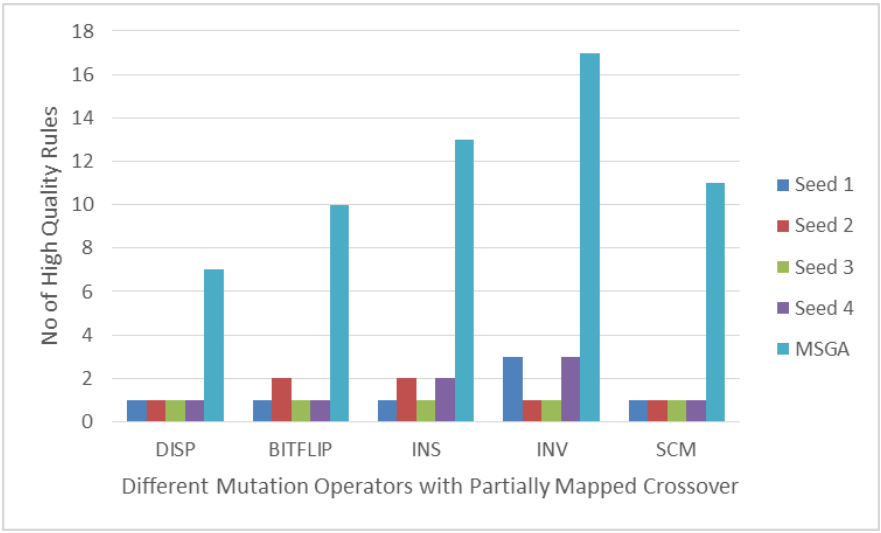


Figure 51: Performance of different seeds and MSGA for different mutation operators with partially mapped crossover for a Monk's Problems data set.

The performance of different seeds and MSGA for different mutation operators with partially mapped crossover for a Monk's Problems data set is shown in Figure 51. According to Figure 51, for all mutation operators the number of generated rules by MSGA is higher than other single seed based genetic algorithms.

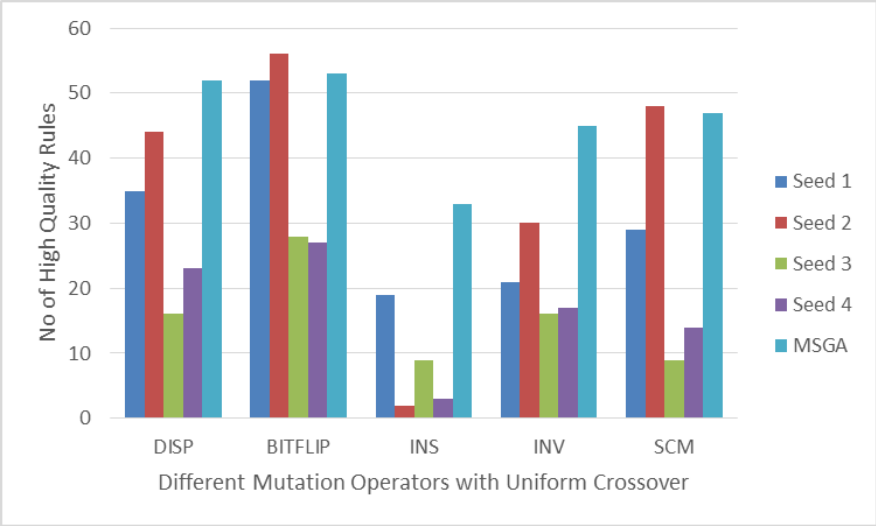


Figure 52: Performance of different seeds and MSGA for different mutation operators with uniform crossover for a Mushroom data set.

The performance of different seeds and MSGA for different mutation operators with uniform crossover for a Mushroom data set is shown in Figure 52. According to Figure 52, for all mutation operators the number of rules generated by MSGA is higher than or similarly to other single seed based genetic algorithms.

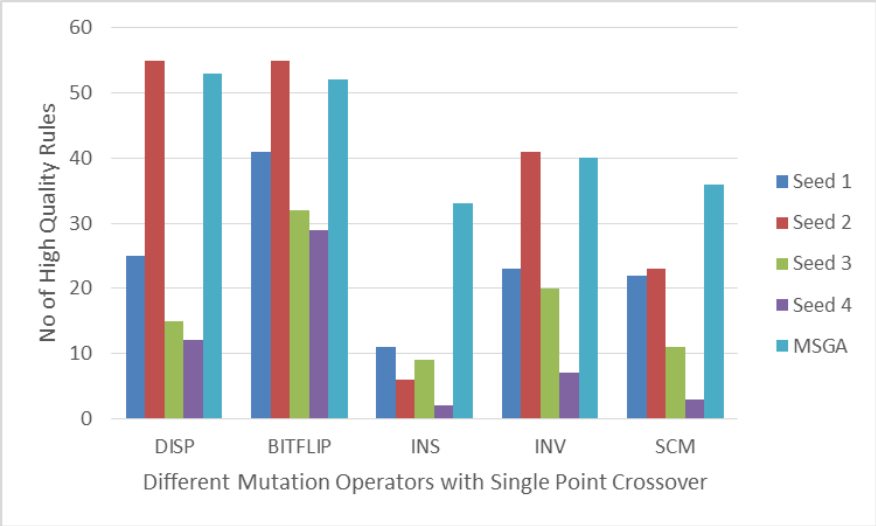


Figure 53: Performance of different seeds and MSGA for different mutation operators with single point crossover for a Mushroom data set.

The performance of different seeds and MSGA for different mutation operators with single point crossover for a Mushroom data set is shown in Figure 53. According to Figure 53, for all mutation operators the number of generated rules by MSGA is higher than or similar to other single seed based genetic algorithms.

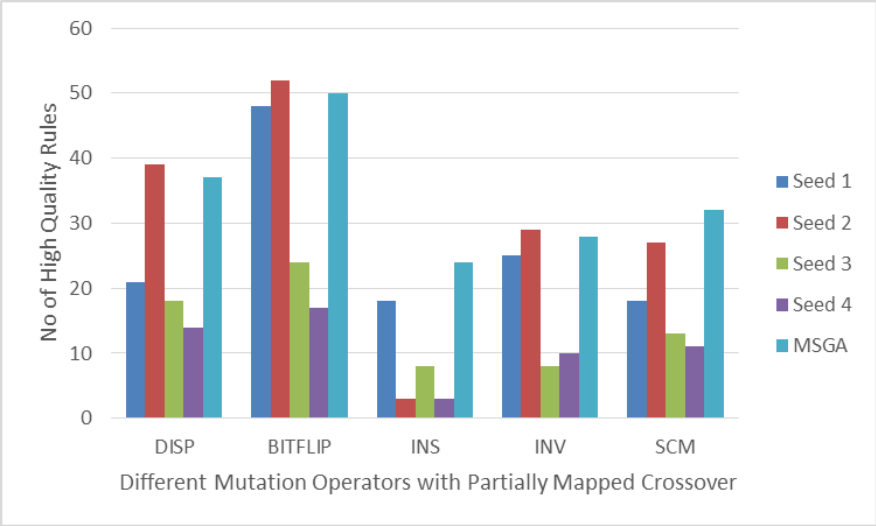


Figure 54: Performance of different seeds and MSGA for different mutation operators with partially mapped crossover for a Mushroom data set.

The performance of different seeds and MSGA for different mutation operators with partially mapped crossover for a Mushroom data set is shown in Figure 54. According to Figure 54, for all mutation operators the number of generated rules by MSGA is higher than or similar to other single seed based genetic algorithms.

5.7.5 Convergence Analysis

Several experiments are carried out for analysing the convergence of the proposed method and single seeds based genetic algorithms for the Breast Cancer data set. The convergence of different single seeds and MSGA using different mutation and crossover operators is shown in Figures 55-57.

The convergence of MSGA and different seeds based genetic algorithms for different mutation operators with uniform crossover for the Breast Cancer data set is shown in Figure 55.

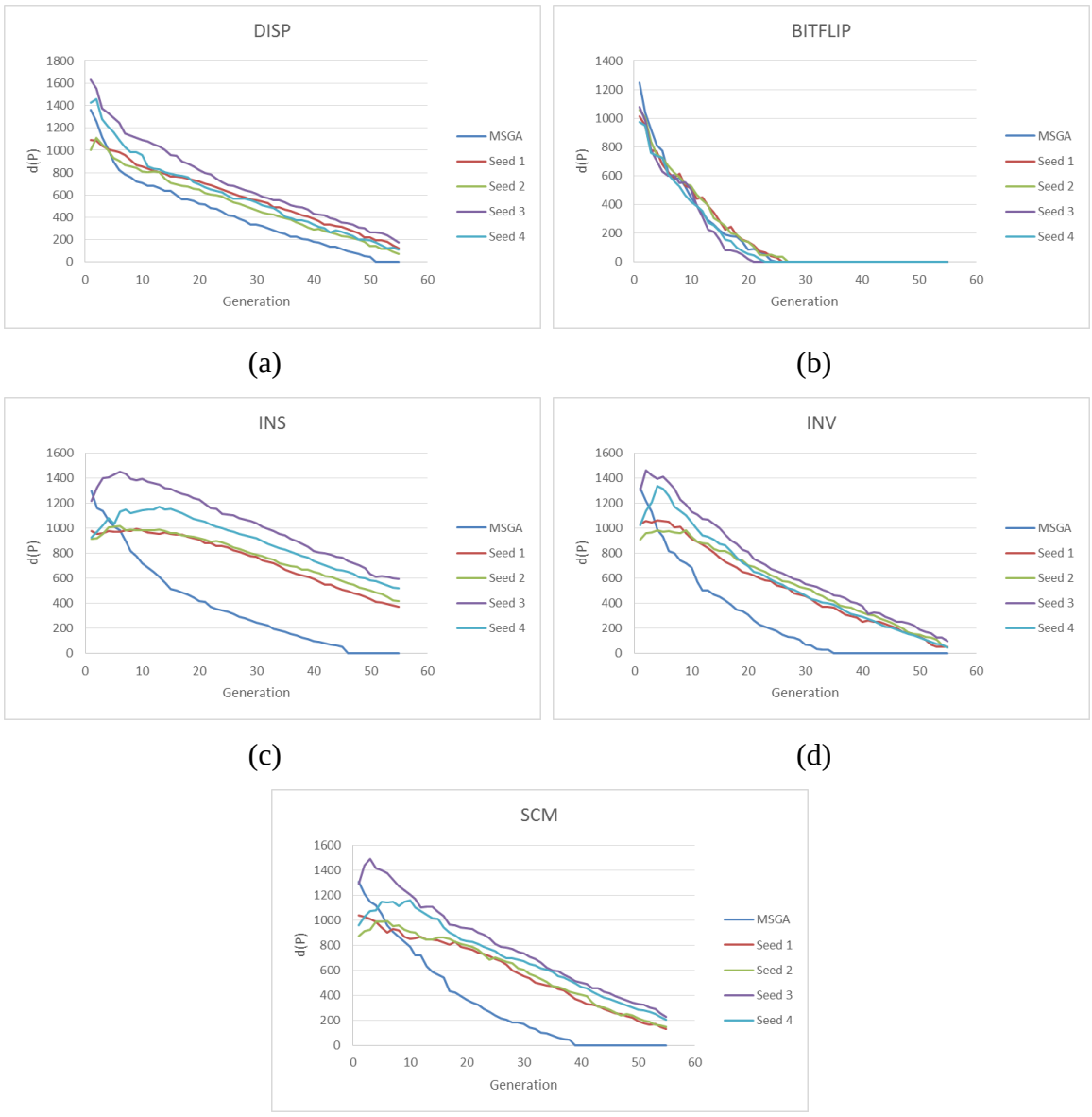


Figure 55: The convergence of MSGA and different seeds based GAs for different mutation operators with uniform crossover for a Breast Cancer data set.

According to Figure 55, all the algorithms scale quite linearly, however in most of the cases MSGA converges faster than single seeds based methods.

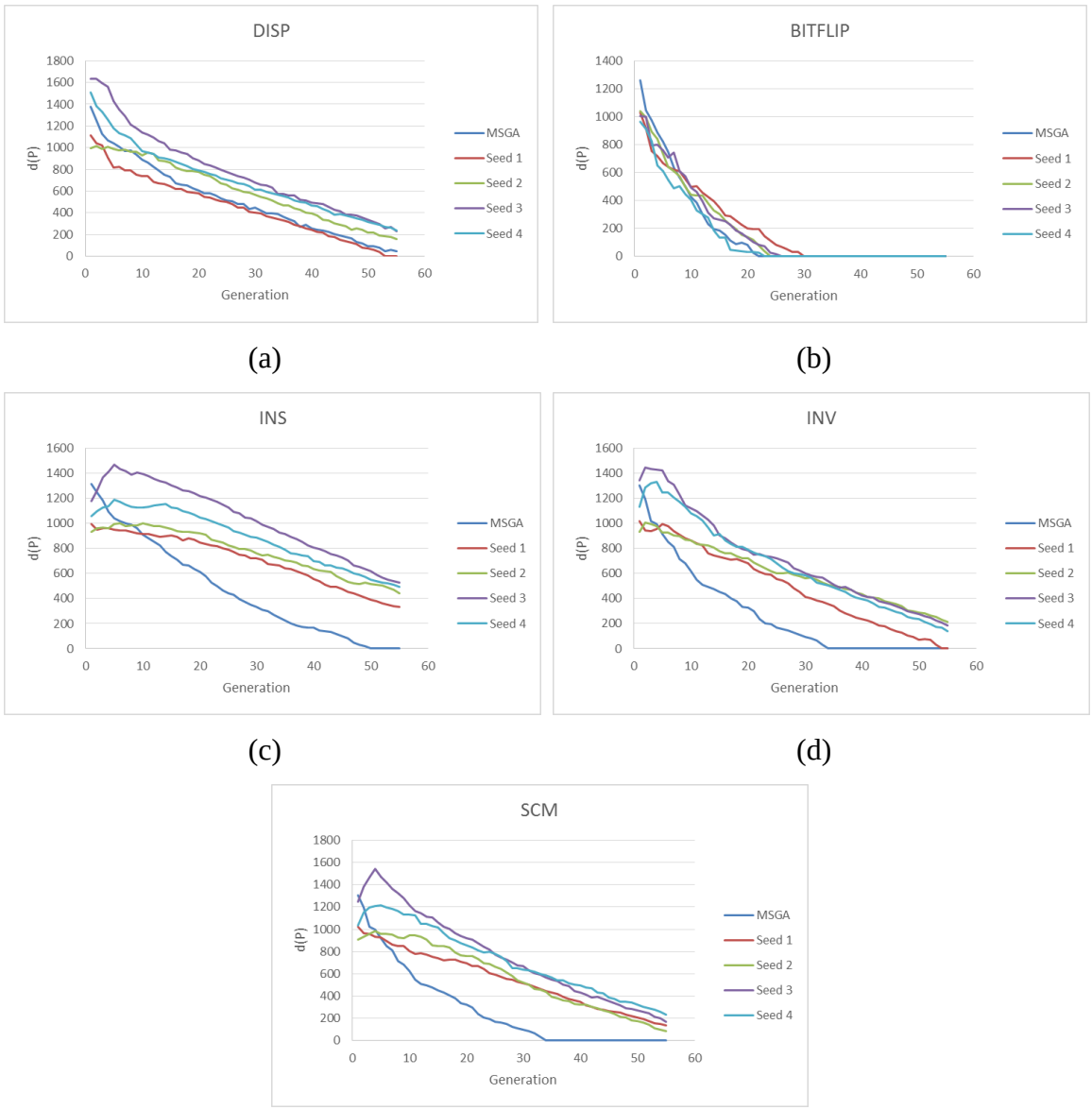


Figure 56: The convergence of MSGA and different seeds based GAs for different mutation operators with single point crossover for a Breast Cancer data set.

The convergence of MSGA and different seeds based genetic algorithms for different mutation operators with single point crossover for the Breast Cancer data set is shown in Figure 56. According to Figure 56, all the algorithms scale quite linearly, however in most of the cases MSGA converges faster than single seeds based methods.

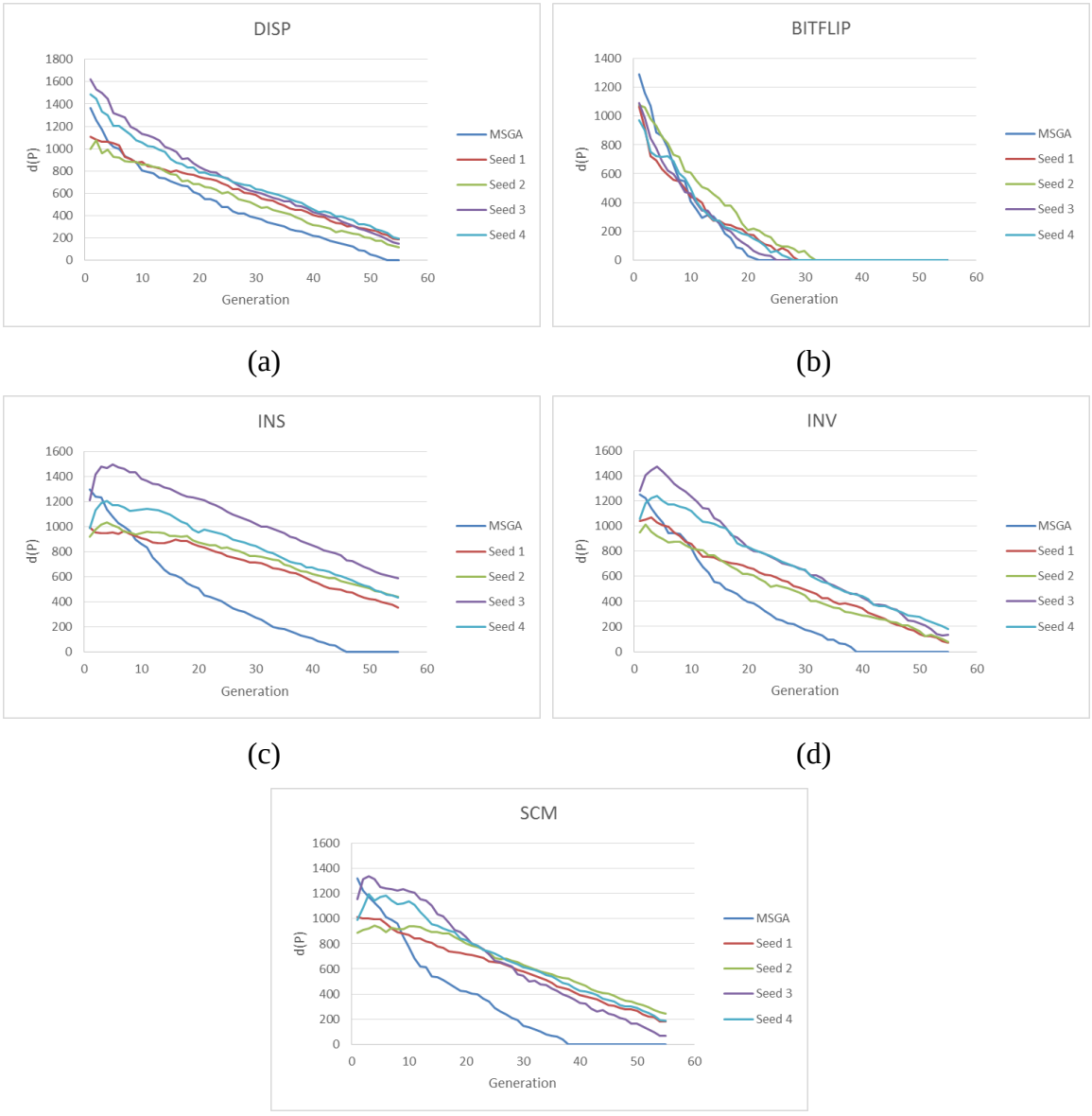


Figure 57: The convergence of MSGA and different seeds based GAs for different mutation operators with partially mapped crossover for a Breast Cancer data set.

The convergence of MSGA and different seeds based genetic algorithms for different mutation operators with single point crossover for the Breast Cancer data set is shown in Figure 57. According to Figure 57, all the algorithms scale quite linearly, however in most of the cases MSGA converges faster than single seeds based methods.

From these figures, it can be concluded that all the algorithms scale quite linearly, however in most of the cases MSGA converges faster than different single seeds based methods.

5.7.6 Conclusion

To discover all high quality rules from a data set and to improve the searching ability and convergence speed of a multiple seeds based genetic algorithm, an m-domain model, m-seeds selection method and m-seeds based initialization approach were introduced to the evolutionary model. Moreover, this study shows the effectiveness and performance of using multiple seeds based approach over single seeds based methods, applying different genetic operators for finding high quality association rules from different data sets.

MSGA was applied successfully with different mutation and crossover operators in mining interesting BARs from categorical data sets and compared the results with different single seeds based genetic algorithms under the same conditions. Taking into account the results obtained over four real world data sets, it can be concluded that the proposed method mines high quality association rules, each having a conditional probability between 80%-100%, with a good trade-off between the convergence speed, search ability and presenting a large number of high quality rules in all the data sets. Moreover, the number of rules for all data sets obtained by the proposed method demonstrates the higher search efficiency when compared with those discovered by single seeds based genetic algorithms.

5.8 Chapter Summary

This chapter showed the effectiveness of the proposed algorithms along with the analysis of the experimental results of each approach. The specifications of the data sets and the parameters which were used for running the methods were also presented in this chapter. Finally, a comparative analysis of the proposed methods with different classical and evolutionary algorithms based approaches was applied to demonstrate the performance of the proposed approaches.

In the next chapter, the summary of findings and the further research works will be discussed.

Chapter 6 - Conclusions

6.1 Introduction

This chapter summarizes how the research work presented in this thesis has obtained its stated goals. The findings are summarized in section 6.2. The contribution and the limitations of this study are discussed in sections 6.3 and 6.4, respectively. The further research directions which are identified during the research work and closing remarks about the effectiveness of evolutionary algorithms for mining interesting association rules are also discussed through section 6.5.

6.2 Summary of Findings

This section outlines the answer to the research questions.

The research question one of this thesis is:

What is required for designing new evolutionary algorithms for mining maximal frequent item sets efficiently?

- This thesis proposes a new approach based on a genetic algorithm to generate maximal frequent item sets (MFIs) from large datasets. This algorithm uses a lexicographic tree and avoids level by level searching which reduces the time required to mine the MFIs in a linear way. The significant contribution of this research is that it generates frequent item sets by the approach based on a genetic algorithm is scale independent to the size of the datasets. The search strategy of this new approach includes bitmap representation of the nodes in a lexicographic tree and identifying frequent item sets (FIs) from superset-subset relationships of nodes. The proposed algorithm shows how the evolutionary method can be used on real datasets to find all the MFIs in an efficient way.
- This genetic based method is improved and extended by another approach named Hybrid GeneticMax, which uses local search along with a genetic algorithm to mine maximal frequent item sets from large data sets. The aim of this new approach is converging to a solution as fast as possible, especially if 1-item sets contain a reasonable amount of infrequent items and the solution resides in the deep level of the lexicographic tree instead of near the root. In addition, a new PSO based approach is developed for discovering the relationship among frequent items along with infrequent ones. Thorough experiments are conducted

for evaluating the performance of newly developed methods. The findings are summarized through the following subsections:

6.2.1 Mining Frequent Patterns Using GeneticMax

In this thesis a new approach (named GeneticMax) is proposed based on a genetic algorithm to mine maximal frequent item sets in an efficient way. Thorough experiments have been conducted on different real data sets. The experimental results demonstrate the following advantages:

- It accesses a large data sets for fewer number of nodes to calculate a support value to find maximal frequent item sets.

- It shows the power of using the evolutionary algorithm for generating frequent item sets from a lexicographic tree. A whole data set is projected onto a lexicographic tree based on a user defined support value.

- The experimental analysis of GeneticMax shows the effect of generations of chromosomes and pruning all the subsets and supersets in both positive and negative boundary areas, which dramatically reduces search space and cost of counting support value of item sets.

- The above advantages of GeneticMax increase the scalability of this algorithm.

A GeneticMax algorithm has been implemented and its performance has been studied. The performance study shows that this algorithm mines different sizes of patterns in real data sets in an efficient way, performs better than other candidate pattern generation and evolutionary based algorithms.

6.2.1.1 Comparative Performance between GeneticMax and Apriori

Several advantages have been demonstrated by the experimental analysis of GeneticMax algorithm in comparison with Apriori algorithm, which are as follows:

- It gives better results than the Apriori algorithm by accessing large data sets for less numbers of nodes especially when the support value is set low by the users.

- For large data sets and low support value, both of these algorithms give the same solution by giving the same number of maximal frequent item sets. To get this solution

Apriori considers a large number of candidate item sets with respect to a genetic based approach.

- For large data sets and high support values, Apriori performs better than a genetic based approach, since the genetic algorithm uses global search mechanism. Apriori uses a level by level search procedure and it gets the solution by accessing less numbers of nodes because the solution is near the root node. The nodes close to the root of lexicographic tree have higher support values.

- Low support value generates a long size frequent pattern which provides information like frequency of an exponential number of smaller sub patterns. In that case a genetic based approach performs better than other existing algorithms.

- The experimental results of a genetic based approach demonstrate the effect of generations of individuals, and prune all the subsets and supersets in a lexicographic tree, which is cost effective in the case of counting the support value and reducing the search space dramatically.

6.2.2 Mining Frequent Patterns Using Hybrid GeneticMax

The Hybrid GeneticMax algorithm used the technique of a local search and a genetic algorithm to mine maximal frequent item sets in an efficient way. Thorough experiments have been carried out on different real data sets for evaluating the performance of the GeneticMax and Hybrid GeneticMax algorithm. The experimental results demonstrate several advantages of the proposed algorithm in comparison with the GeneticMax algorithm.

1) It considers fewer item sets of large data sets to calculate support value to get the solution i.e. finding maximal frequent item sets.

2) It shows the power of using an evolutionary algorithm along with a local search mechanism for generating frequent item sets from lexicographic trees. Abstract representation of large data sets is done by a lexicographic tree based on a user defined support value, which is used as a search space for these experiments.

3) The experimental analysis of the Hybrid GeneticMax shows the effect of a local search along with a global search mechanism, and it compared the results with the GeneticMax algorithm. Firstly, it sorted out the infrequent items from 1- item sets using a

local search mechanism and it used these infrequent item sets for further pruning methodologies. After this step, it used a global search mechanism i.e. using a genetic based approach it prunes all the subsets and supersets in both positive and negative boundary areas, which dramatically reduces search space and the cost of counting the support value of item sets.

4) The above advantages of the Hybrid GeneticMax increases the searching speed and scalability of this algorithm which is shown through comparative analysis of GeneticMax and the Hybrid GeneticMax algorithm.

5) This approach outperformed the GeneticMax algorithm, if there are a reasonable amount of infrequent items in 1-item sets. For a certain threshold value, if there are no infrequent items in 1-item sets, then this approach performs similar to the GeneticMax algorithm.

6.2.3 Mining Association Rules for Both Frequent and Infrequent Items Using PSO

To mine association rules for both frequent and infrequent item sets in an efficient way, in this thesis another approach is proposed which is based on the Particle Swarm Optimization Algorithm. The experimental results demonstrate several advantages of the proposed method in comparison with other existing algorithms.

1) It generates frequent and infrequent item sets near the cut in the lexicographic tree.

2) Previous researchers also applied PSO for association rule mining, however, their studies showed the mining results for only two or three items. This approach can mine association rules for more than three items if it satisfies user defined threshold confidence values. In addition, this approach considers user define threshold values which helps users mine those rules which are interesting to them.

2) It shows the power of using a heuristic algorithm for generating association rules for frequent item sets along with infrequent ones from a lexicographic tree.

3) The experimental analysis of this approach shows the effect of generations of particles in a search space and pruning all the subsets and supersets in both positive and negative boundary areas, which dramatically reduces search space and cost of counting support value of item sets.

The research question two of this thesis is:

Which mechanisms are used for designing new multi-objective evolutionary algorithms for discovering a reduced set of high quality Boolean association rules?

- A new multi-objective evolutionary model named Association Rules Mining with Genetic Algorithm Using an Adaptive Mutation Method (ARMGAAM), which is very useful for mining reduced sets of Boolean association rules from categorical data sets. Another method named Mining Boolean Association Rules with Evolutionary Algorithm (MBAREA), a new evolutionary model which extends the existing Association Rule Mining with Genetic Algorithm (ARMGA) and Multi-objective Association Rule Mining with Genetic Algorithm (ARMMGA), maximizes two objectives; performance and interestingness. The former method uses a re-initialization technique along with an adaptive mutation method whereas the latter uses a class based mutation method along with a best population technique. Thorough experiments are conducted for evaluating the performance of newly developed methods. The findings are summarized through the following subsections:

6.2.4 Mining Interesting Association Rules Using ARMGAAM

ARMGAAM is used to mine a reduced set of Boolean association rules. The generated BARs are interesting, easy to understand and maximizing three objectives: lift, net confidence and conditional probability. To accomplish this, this algorithm extends the ARMGA algorithm for performing evolutionary learning and selection of a condition of each rule. This proposed approach introduces the re-initialization process and the adaptive mutation method to its evolutionary model in order to perform evolutionary learning and to improve the diversity of the obtained set of rules. Moreover, the obtained rules are strong, showing a strong relationship among the item sets and solving the problem of the support-confidence framework.

From the experimental results obtained over four real world data sets, it can be concluded that the proposed approach allows users to mine a reduced set of BARs with good trade- off among the number of generated rules, support, confidence, lift, net confidence and conditional probability of all the data sets. Finally, the proposed approach has a good computational cost and scalability when the problem size increases.

6.2.5 Mining Interesting Association Rules Using MBAREA

Another approach has been proposed, named MBAREA, a new EA for mining a reduced set of positive BARs. The generated rules are interesting, easy to understand and maximize two objectives performance and interestingness. To accomplish this, this approach extends the existing ARMGA and ARMMGA for evolutionary learning and selection of a condition of each rule. This algorithm introduces class based mutation method to the evolutionary model and a best population technique to improve the diversity of the generated rules and to store all the non-dominated rules which are generated in the intermediate generation of a population. Analyzing the results obtained over six real world data sets, it can be concluded that the generated rules maintain the good trade-off among the number of rules, confidence, conditional probability, interest and lift values in all the data sets. Moreover, the generated rules are very strong which indicates a strong relationship between the item sets and solves the drawback of support dependent methods. Finally, the experimental results show that the proposed algorithm has a good computational cost and scales well when the problem size is increased.

The research question three of this thesis is:

What are the techniques by which an effective initial population is generated for further evolution based on multiple seeds?

- Most of the association rule mining algorithms which are based on GA, use a single seed chromosome for generating an initial set of solutions. In this thesis, a new model is developed which generates multiple seeds from multiple domains of a solution space and an initial population is generated based on those seeds. The comparative analysis of this newly developed method with different single seed based algorithms with respect to different mutation and crossover operators demonstrate the efficiency of the proposed approach. The findings are summarized through the following subsections:

6.2.6 Effects of A Multiple Seeds Based Genetic Algorithm on Discovering Association Rules

Multiple seeds based genetic algorithm generates multiple seeds from an m- domain solution space for producing an effective initial population for further evolutionary learning to mine a large number of high quality association rules from categorical data

sets. This approach introduces an *m-domain* model, *m-seeds* selection method and *m-seeds* based initialization approach to the evolutionary model in order to discover all the high quality rules and to improve the searching ability and convergence speed. Moreover, this study shows the effectiveness and performance of using a multiple seeds based approach over single seeds based methods, applying different genetic operators for finding high quality association rules from different data sets.

MSGA is successfully applied with different mutation and crossover operators in mining interesting BARs from categorical data sets and compared the results with different single seeds based genetic algorithms under the same conditions. Taking into account the results obtained over four real world data sets, it can be concluded that the proposed method mines high quality association rules, each having a conditional probability between 80%-100%, with a good trade-off among the convergence speed, search ability and presenting a large number of high quality rules in all the data sets. Moreover, the number of rules for all data sets obtained by the proposed method shows the higher search efficiency than single seeds based genetic algorithms.

6.3 Contribution

This thesis proposes a new method named GeneticMax, based on a genetic algorithm, which is used to mine maximal frequent item sets by accessing a large data set for fewer number of nodes. This algorithm uses a lexicographic tree and avoids level by level searching which reduces the time required to mine the MFIs in a linear way. The significant contribution of this research is that it generates frequent item sets by the approach based on a genetic algorithm is scale independent to the size of the datasets. This method is improved by another approach named Hybrid GeneticMax. This new model which outperforms the GeneticMax algorithm if there are a reasonable amount of infrequent items in 1- item sets. This proposal shows the power of using an evolutionary algorithm along with a local search mechanism for generating maximal frequent item sets from a lexicographic tree. On the other hand, this research proposed PSO based approach, a new heuristic algorithm for mining association rules for both frequent and infrequent items. This approach can mine rules for more than three items.

This study also proposes a new multi-objective evolutionary model named Association Rules Mining with Genetic Algorithm Using an Adaptive Mutation Method (ARMGAAM), which is very useful for mining reduced sets of Boolean association

rules from categorical data sets. Based on the data sets and design factors, another method is introduced in this thesis named Mining Boolean Association Rules with Evolutionary Algorithm (MBAREA). This is a new evolutionary model which extends the existing Association Rule Mining with Genetic Algorithm (ARMGA) and Multi-objective Association Rule Mining with Genetic Algorithm (ARMMGA). This approach maximizes two objectives; performance and interestingness. The former method uses a re-initialization technique along with an adaptive mutation method whereas the latter uses a class based mutation method along with a best population technique. Both methods extracts a reduced set of BARs from different data sets with a good trade-off among the number of generated rules and different measures.

Finally, a new genetic algorithm based on multiple seeds is proposed for producing an effective initial population. Experimental results demonstrate that this approach has a higher search efficiency along with good convergence speed, prevents the limitation of selecting an effective single seed for generating an initial population for mining BARs. The selection of above mentioned evolutionary algorithms depends on the specific needs of users.

Thorough experiments are conducted on well-known real world data sets such as Breast Cancer, Solar Flare, Mushroom, and so on for evaluating the performance of newly developed methods and compared the results with existing classical and evolutionary based approaches.

6.4 Limitations of the Study

In this thesis, different evolutionary algorithms are presented for mining maximal frequent item sets and Boolean association rules. To mine Boolean association rules this study only considers categorical data sets. In real world applications, data sets not only use categorical values but also contain quantitative or numeric values for mining quantitative and fuzzy association rules. For this reason, these algorithms need to modify in such a way that it could apply on those data sets which contain quantitative or numeric values for mining quantitative and fuzzy association rules.

6.5 Future Research

In this section, some future works, identified during the thesis, are discussed.

6.5.1 Adapting the Proposed Methods for Other Data Mining Techniques

In this thesis, different evolutionary algorithms have been presented for discovering frequent patterns and association rules. The adaptation of these approaches could be very helpful for other data mining techniques such as web mining and associative classification (Thabtah 2007).

Web mining is the process of discovering knowledge and patterns from the web (Kosala & Blockeel 2000). Web content mining, web structure mining, and web usage mining are three different types of web mining. The aim of web content mining is to search information within web pages, whereas web structures mining is focused on the hyperlink structures of the web. Web usage mining is focused on the behaviour of users when they interact with the web (Srivastava et al. 2000). Several association rule mining algorithms are proposed to discover pages that are often visited by the users can reveal a group of users (Eirinaki & Vazirgiannis 2003). Web mining is an emerging field that introduces many problems such as grouping the users based on the same set of pages they are often visited, page association based on the pages that are browsed together, or finding browse and navigation orders which are followed by many users. These problems could be solved by adapting the proposed methods in web mining.

On the other hand, associative classifiers or associative classification mining is another emerging field in data mining research area which uses association rule discovery methods to model classification systems. In the last few years, few algorithms have been proposed by the researchers which employ several methods such as rule discovery, rule pruning and rule evaluation. Association rule mining and classification are the two data mining tasks which are integrated by the Associative classification for building a model for the purpose of prediction. Association rule mining and classification are the similar tasks in data mining, with the difference that association rule mining describes correlation among the items in a large data set whereas the main aim of classification is to predict the class labels (Thabtah 2007; Liu et al. 1998).

Both techniques are essential in real world applications of data mining. So the integration of association rule mining along with classification may be of great interest to the users.

6.5.2 Adapting the Proposed Methods for Different Metrics

For mining quantitative or fuzzy association rules, researchers use different metrics to measure the quality of a rule. These metrics which are optimised by different approaches in the multi-objective framework include a lift, coverage, comprehensibility, surprise, recall, cosine, and so on. Based on the performance of the algorithms and data sets, different metrics are chosen by different researchers as their objective functions. Different metrics from different groups are selected by the researchers to make the objective functions uncorrelated and contradictory (Khabzaoui et al. 2008). However, a systemic comparative analysis among the chosen metrics could be of great interest to the users (Mukhopadhyay et al. 2014). Therefore, taking into account the different metrics, the proposed methods must be adapted in such a way that it will classify the metrics into different domains such as conflicting, non-conflicting, correlated, and so on.

6.5.3 Designing New Evolutionary Algorithms for Mining Association Rules for Problems with Special Features

The progress made on the development of the evolutionary algorithms for discovering association rules allows the focus on further studies on data mining problems with special features such as data sets containing attributes with missing values, low quality and large volumes of data.

In data mining communities, most of the association rule mining algorithms have been focused on complete records or data sets with accurate values. However, in real world applications many data sets have a certain degree of imprecision, i.e. missing records, uncertain and low quality data. Several algorithms have been proposed which omit missing records. Ignoring missing values cannot be the ideal solution since unknown values may contain important information (Thabtah 2007). Therefore, designing new algorithms which are able to deal with the uncertain data and efficiently handle and exploit the information contained in the data sets with missing and low quality values, represent a challenging issue for the future researchers (Palacios et al. 2011; Thabtah 2007).

Moreover, the storage and generation of large volumes of data further expand the process of analysing and extracting the knowledge from large data sets with the belief that the resulting information may be accurate depends on the availability of more data (Sathi 2012, pp. 15-46). However, traditional algorithms which are used in data mining are often not able to deal with large data sets. Therefore, redesign and adaptation for association rule mining algorithms are necessary for handling large volumes of data and maintaining the quality of the obtained set of rules.

Chapter 7 - Bibliography

Agarwal, R.C., Aggarwal, C.C. & Prasad, V.V. V, 2001. A tree projection algorithm for generation of frequent itemsets. *Parallel and Distributed Computing- Special Issue on High Performance Data Mining*, 61(3), pp.350–371.

Agarwal, R.C., Aggarwal, C.C. & Prasad, V.V. V., 2000. Depth first generation of long patterns. In *6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. pp. 108–118.

Aggarwal, C.C. & Yu, P.S., 1998. A new framework for itemset generation. In *17th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*. pp. 18–24.

Agrawal, R., Imieliński, T. & Swami, A., 1993. Mining association rules between sets of items in large databases. In *ACM SIGMOD International Conference on Management of Data*. pp. 207–216.

Agrawal, R. & Shafer, J.C., 1996. Parallel mining of association rules. *IEEE Transactions on Knowledge and Data Engineering*, 8(6), pp.962–969.

Agrawal, R. & Srikant, R., 1994. Fast algorithms for mining association rules. In *20th International Conference on Very Large Data Bases*. pp. 487–499.

Agrawal, R. & Srikant, R., 1995. Mining sequential patterns. In *Eleventh International Conference on Data Engineering*. pp. 3–14.

Ahn, K.-I. & Kim, J.-Y., 2004. Efficient mining of frequent itemsets and a measure of interest for association rule mining. *Journal of Information and Knowledgement*, 3(3), pp.245–257.

Alatas, B. & Akin, E., 2008a. MODENAR: Multi-objective differential evolution algorithm for mining numeric association rules. *Applied Soft Computing*, 8(1), pp.646–656.

Alatas, B. & Akin, E., 2008b. Rough particle swarm optimization and its applications in data mining. *Soft Computing*, 12(12), pp.1205–1218.

Alataş, B. & Akin, E., 2005. An efficient genetic algorithm for automated mining of both positive and negative quantitative association rules. *Soft Computing*, 10(3), pp.230–237.

Albayrak, M. & Allahverdi, N., 2011. Development a new mutation operator to solve the Traveling Salesman Problem by aid of Genetic Algorithms. *Expert Systems with Applications*, 38(3), pp.1313–1320.

Anon, 2014. Genetic Algorithms. Available at: http://en.wikipedia.org/wiki/Genetic_algorithm.

Avci, E., Abdulkadir, S. & Davut, H., 2009. An optimum feature extraction method for texture classification. *Expert Systems with Applications*, 36(3), pp.6036–6043.

Banzhaf, W., 1990. The “Molecular” traveling salesman. *Biological Cybernetics*, 64(1), pp.7–14.

Bayardo, R.J., 1998. Efficiently mining long patterns from databases. In *ACM SIGMOD International Conference on Management of Data*. pp. 85–93.

Beasley, D., Bull, D.R. & Martin, R.R., 1993. An overview of genetic algorithms : Part

1 , fundamentals. *University of computing*, 15(2), pp.58–69.

Berzal, F., Blanco, I. & S, D., 2002. Measuring the accuracy and interest of association rules : A new framework. *Intelligent Data Analysis*, 6(3), pp.221–235.

Bhanu, B. & Lin, Y., 2003. Genetic algorithm based feature selection for target detection in SAR images. *Image and Vision Computing*, 21(7), pp.591–608.

Borgelt, C., 2003. Efficient implementations of Apriori and Eclat. In *IEEE ICDM Workshop on Frequent Item Set Mining Implementations*. pp. 280–296.

Borgelt, C., 2012. Frequent item set mining. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 2(6), pp.437–456.

Burdick, D. et al., 2005. MAFIA: A maximal frequent itemset algorithm. *IEEE Transactions on Knowledge and Data Engineering*, 17(11), pp.1490–1504.

Burdick, D., Calimlim, M. & Gehrke, J., 2005. MAFIA: A maximal frequent itemset algorithm for transactional databases. In *17th International Conference on Data Engineering*. pp. 443–452.

Cameron, J.J. & Leung, C.K., 2011. Mining Frequent Patterns from Precise and Uncertain Data. *Computing and system*, 1(1), pp.3–22.

Chang, P.C., Huang, W.H. & Ting, C.J., 2010. Dynamic diversity control in genetic algorithm for mining unsearched solution space in TSP problems. *Expert Systems with Applications*, 37(3), pp.1863–1878.

Chen, F., Hang, J. & Zhang, Q., 2006. An efficiently algorithm for mining association rules. In *IKE*. pp. 155–161.

Chen, X., 2003. An improved branch and bound algorithm for feature selection. *Pattern Recognition Letters*, 24(12), pp.1925–1933.

Cho, H. et al., 2008. Genetic algorithm-based feature selection in high-resolution NMR spectra. *Expert Systems with Applications*, 35(3), pp.967–975.

Clerc, M., 1999. The swarm and the queen: towards a deterministic and adaptive particle swarm optimization. In *The Congress of Evolutionary Computation*. pp. 1951–1957.

Cohen, E. et al., 2001. Finding Interesting Association without Support Pruning. *IEEE Transactions on Knowledge and Data Engineering*, 13(1), pp.64–78.

Dou, W. et al., 2008. Quick response data mining model using genetic algorithm. In *SICE Annual Conference*. pp. 1214–1219.

Du, F. et al., 2009. Mining gene network by combined association rules and genetic algorithm. In *International Conference on Communications, Circuits and Systems*. Milpitas, CA, pp. 581–585.

Eberhart, R.C. & Shi, Y., 2001. Particle Swarm Optimization : developments , applications and resources. In *The Congress on Evolutionary Computation*. pp. 81–86.

Eirinaki, M. & Vazirgiannis, M., 2003. Web mining for web personalization. *ACM Trans.Inter.Tech.*, 3(1), pp.1–27.

Fayyad, U., Piatetsky-Shapiro, G. & Smyth, P., 1996. From data mining to knowledge discovery in databases. *Advances in knowledge discovery and data mining*, 17(3), pp.37–54.

Fogel, D.B., 1988. An evolutionary approcah to the Traveling Sales Problem. *Biological Cybernetics*, 60(2), pp.139–144.

Fogel, D.B., 1993. Empirical estimation of the computation required to discover approximate solutions to the Traveling Salesman Problem using evolutionary programming. In *2nd Annual Conference on Evolutionary Programming*. pp. 56–61.

Freitas, A.A., 2003. A survey of evolutionary algorithms for data mining and knowledge discovery. In *Advances in Evolutionary Computing*. pp. 819–845.

Geng, L. & Hamilton, H.J., 2006. Interestingness measures for data mining. *ACM Computing Surveys*, 38(3), pp.1–32.

Ghosh, A. & Nath, B., 2004. Multi-objective rule mining using genetic algorithms. *Information Sciences*, 163(1-3), pp.123–133.

Ghosh, S. et al., 2012. Association rule mining algorithms and genetic algorithm: A comparative study. In *3rd International Conference on Emerging Applications of Information Technology*. pp. 202–205.

Ghosh, S. et al., 2011. Weather data mining using artificial neural network. In *IEEE Recent Advances in Intelligent Computational Systems*. pp. 192–195.

Gouda, K. & Zaki, M.J., 2005. GenMax : An efficient algorithm for mining maximal frequent itemsets. *Data Mining and Knowledge Discovery*, 11(3), pp.223–242.

Grefenstette, J. et al., 1985. Genetic algorithms for the TSP. In *First International Conference on Genetic Algorithms and Their Applications*. pp. 160–165.

Gunal, S. et al., 2009. The search for optimal feature set in power quality event classification. *Expert Systems with Applications*, 36(7), pp.10266–10273.

Han, J. et al., 2007. Frequent pattern mining: current status and future directions. *Data Mining and Knowledge Discovery*, 15(1), pp.55–86.

Han, J. & Kamber, M., 2006. *Data Mining: Concepts and Techniques* 2nd ed., Burlington, MA, USA: Morgan Kaufmann.

Han, J., Pei, J. & Yin, Y., 2000. Mining frequent patterns without candidate generation. In *ACM SIGMOD International Conference on Management of Data*. pp. 1–12.

Helsgaun, K., 2000. An effective implementation of the Lin–Kernighan traveling salesman heuristic. *European Journal of Operational Research*, 126(1), pp.106–130.

Hipp, J., Güntzer, U. & Nakhaeizadeh, G., 2000. Algorithms for association rule mining—a general survey and comparison. *ACM SIGKDD Explorations*, 2(1), pp.58–64.

Hong, T. et al., 2008. Genetic-fuzzy data mining With Divide-and-Conquer strategy. *IEEE Transactions on Evolutionary Computation*, 12(2), pp.252–265.

Huang, J., Che-tsung, Y. & Fu, C., 2004. A genetic algorithm based searching of maximal frequent itemsets. In *International Conference on Artificial Intelligence*. pp. 548–554.

Jayalakshmi, G.A., Sathiamoorthy, S. & Rajaram, R., 2001. A hybrid genetic algorithm-a new approach to solve Travelling Salesman Problem. *International Journal of Computational Engineering Science*, 2(2), pp.339–355.

Jesus, M.J. del et al., 2011. On the discovery of association rules by means of

evolutionary algorithms. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 1(5), pp.397–415.

Kabir, M.M.J. et al., 2015a. A new evolutionary algorithm for extracting a reduced set of interesting association rules. In *22nd International Conference On Neural Information Processing*. pp. 133–142.

Kabir, M.M.J. et al., 2014. A novel approach to mining maximal frequent itemsets based on genetic algorithm. In *9th International Conference on Information Technology and Applications*. Sydney, Australia, pp. 1–6.

Kabir, M.M.J. et al., 2015b. Comparative analysis of genetic based approach and apriori algorithm for mining maximal frequent item sets. In *IEEE Congress on Evolutionary Computation*. pp. 39–45.

Kannimuthu, S. & Premalatha, K., 2014. Discovery of high utility itemsets using genetic algorithm with ranked mutation. *Applied Artificial Intelligence*, 28(4), pp.337–359.

Kantardzic, M., 2003. *Data Mining: Concepts, Models, Methods and Algorithms*, John Wiley & Sons, Inc.

Kaya, İ., 2009. A genetic algorithm approach to determine the sample size for control charts with variables and attributes. *Expert Systems with Applications*, 36(5), pp.8719–8734.

Kaya, M. & Alhajj, R., 2005. Genetic algorithm based framework for mining fuzzy association rules. *Expert Systems with Applications*, 152(3), pp.587–601.

Kaya, M. & Alhajj, R., 2006. Utilizing genetic algorithms to optimize membership functions for fuzzy weighted association rules mining. *Applied Intelligence*, 24(1), pp.7–15.

Khabzaoui, M., Dhaenens, C. & Talbi, E.-G., 2008. Combining evolutionary algorithms and exact approaches for multi-objective knowledge discovery. *RAIRO Operations Research*, 42(1), pp.69–83.

Khan, A., Bawane, N.G. & Bodkhe, S., 2010. An analysis of particle swarm optimization with data clustering-technique for optimization in data mining. *International Journal on Computer Science and Engineering*, 2(4), pp.1363–1366.

Kosala, R. & Blockeel, H., 2000. Web mining research: A survey. *SIGKDD Explorations*, 2(1), pp.1–15.

Kudo, M. & Sklansky, J., 2000. Comparison of algorithms that select features for pattern classifiers. *Pattern Recognition*, 33(1), pp.25–41.

Kuo, R., Chao, C. & Chiu, Y., 2011. Application of particle swarm optimization to association rule mining. *Applied Soft Computing*, 11(1), pp.326–336.

Kuo, R.J. & Shih, C.W., 2007. Association rule mining through the ant colony system for national health insurance research database in Taiwan. *Computers & Mathematics with Applications*, 54(11-12), pp.1303–1318.

Lin, D.I. & Kedem, Z.M., 2002. Pincer-search: An efficient algorithm for discovering the maximum frequent set. *IEEE Transactions on Knowledge and Data Engineering*, 14(3), pp.553–566.

Lin, J.-Y. et al., 2008. Classifier design with feature selection and feature extraction using layered genetic programming. *Expert Systems with Applications*, 34(2),

pp.1384–1393.

Liu, B. et al., 2000. Analyzing the subjective interestingness of association rules. *IEEE Intelligent Systems*, 15(5), pp.47–55.

Liu, B. et al., 1998. Integrating classification and association rule mining. In *4th International Conference on Knowledge Discovery and Data Mining*. pp. 80–86.

Lu, H. et al., 1996. Effective data mining using neural networks. *IEEE Transactions on Knowledge and Data Engineering*, 8(6), pp.957–961.

Maaranen, H., Miettinen, K. & Mäkelä, M.M., 2004. Quasi-Random initial population for genetic algorithms. *Computers and Mathematics with Applications*, 47(12), pp.1885–1895.

Maaranen, H., Miettinen, K. & Penttinen, A., 2007. On initial populations of a genetic algorithm for continuous optimization problems. *Journal of Global Optimization*, 37(3), pp.405–436.

Mabroukeh, N.R. & Ezeife, C.I., 2010. A taxonomy of sequential pattern mining algorithms. *ACM Computing Surveys*, 43(1), pp.1–41.

Man, K.F., Tang, K.S. & Kwong, S., 1996. Genetic algorithms : concepts and applications. *IEEE Transactions on Industrial Electronics*, 43(5), pp.519–534.

Martin, D. et al., 2014. A new multiobjective evolutionary algorithm for mining a reduced set of interesting positive and negative quantitative association rules. *IEEE Transactions on Evolutionary Computation*, 18(1), pp.54–69.

Mei, Q. et al., 2006. Generating semantic annotations for frequent patterns with context analysis. In *12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. pp. 337–346.

Merwe, D.W. van der & Engelbrecht, A., 2003. Data clustering using particle swarm optimization. In *The 2003 Congress on Evolutionary Computation*. pp. 215–220.

Michalewicz, Z., 1992. *Genetic algorithms + data structures = evolution programs*, Berlin: Springer.

Mukhopadhyay, A. et al., 2014. A survey of multiobjective evolutionary algorithms for data mining: Part I. *IEEE Transactions on Evolutionary Computation*, 18(1), pp.4–19.

Palacios, A.M., Sánchez, L. & Couso, I., 2011. Future performance modeling in athletism with low quality data-based genetic fuzzy systems. *Journal of Multiple-Valued Logic and Soft Computing*, 17(2-3), pp.207–228.

Pears, R. & Koh, Y.S., 2012. Weighted Association Rule Mining Using Particle Swarm Optimization. In *New Frontiers in Applied Data Mining*. Springer-Verlag Berlin Heidelberg, pp. 327–338.

Piatetsky-Shapiro, G., 1991. Discovery, analysis, and presentation of strong rules. In *Knowledge Discovery in Databases*. pp. 229–248.

Premalatha, K. & Natarajan, A., 2009. Genetic algorithm for document clustering with simultaneous and ranked mutation. *Modern Applied Science*, 3(2), pp.75–82.

Qodmanan, H.R., Nasiri, M. & Minaei-Bidgoli, B., 2011. Multi objective association rule mining with genetic algorithm without specifying minimum support and minimum confidence. *Expert Systems with Applications*, 38(1), pp.288–298.

Ramaswamy, S., Mahajan, S. & Silberschatz, A., 1998. On the discovery of interesting

patterns in association rules. In *24th International conference on very large data bases*. pp. 368–379.

Russell, S. & Norvig, P., 2008. *Artificial Intelligence: A Modern Approach*, Prentice Hall Series in Artificial Intelligence.

Salleb, A., Maazouzi, Z. & Vrain, C., 2002. Mining maximal frequent itemsets by a Boolean based approach. In *European Conference on Artificial intelligence*. pp. 285–289.

Salleb-aouissi, A. et al., 2013. QuantMiner for mining quantitative association rules. *Machine Learning Research*, 14(1), pp.3153–3157.

Salleb-aouissi, A., Vrain, C. & Nortet, C., 2007. QuantMiner : A genetic algorithm for mining quantitative association rules. In *20th International Joint Conference on Artificial Intelligence*. pp. 1035–1040.

Sathi, A., 2012. *Big data analytics: disruptive technologies for changing the game*, MC Press.

Seo, D. & Moon, B., 2002. Voronoi quantized crossover for Traveling Salesman Problem. In *Genetic and Evolutionary Computation Conference*. pp. 544–552.

Shenoy, P. et al., 2005. Dynamic association rule mining using genetic algorithms. *Intelligent Data Analysis*, 9(5), pp.439–453.

Shenoy, P. et al., 2003. Evolutionary approach for mining association rules on dynamic databases. In *7th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining*. pp. 325–336.

Shi, Y. & Eberhart, R., 1998. A modified particle swarm optimizer. In *IEEE International Conference on Evolutionary Computation*. pp. 69–73.

Silverstein, C., Brin, S. & Motwani, R., 1998. Beyond market baskets : generalizing association rules to dependence rules. *Data Mining and Knowledge Discovery*, 2(1), pp.39–68.

Snyder, W.C., 2000. Accuracy estimation for quasi-Monte Carlo simulations. *Mathematics and Computers in Simulation*, 54(1-3), pp.131–143.

Song, W., Li, C.H. & Park, S.C., 2009. Genetic algorithm for text clustering using ontology and evaluating the validity of various semantic similarity measures. *Expert Systems with Applications*, 36(5), pp.9095–9104.

Srinivas, M. & Patnaik, L.M., 1994. Genetic algorithms: A survey. *Computer*, 27(6), pp.17–26.

Srivastava, J. et al., 2000. Web usage mining: discovery and applications of usage patterns from web data. *ACM SIGKDD Explorations Newsletter*, 1(2), pp.12–23.

Stutzle, T. & Hoos, H., 1997. Max-Min ANT System and local search for the traveling salesman problem. In *IEEE International Conference on Evolutionary Computation*. pp. 309–314.

Thabtah, F., 2007. A review of associative classification mining. *The Knowledge Engineering Review*, 22(1), pp.37–65.

Toivonen, H., 1996. Sampling large databases for association. In *22nd International Conference on Very Large Data Bases*. pp. 134–145.

Uncu, Ö. & Türkşen, I.B., 2007. A novel feature selection approach: combining feature wrappers and filters. *Information Sciences*, 177(2), pp.449–466.

Wakabi-Waiswa, P.P. & Baryamureeba, V., 2008. Extraction of interesting association rules using genetic algorithms. *International Journal of Computing and ICT Research*, 2(1), pp.101–110.

Wan, W. & Birch, J.B., 2013. An improved hybrid genetic algorithm with a new local search procedure. *Journal of Applied Mathematics*, 2013, pp.1–10.

Wang, H. et al., 2003. A maximal frequent itemset algorithm. In *9th International Conference on Rough Sets, Fuzzy Sets, Data Mining and Granular Computing*. Chongqing, China: Springer Berlin Heidelberg, pp. 484–490.

Wang, K. et al., 2001. Mining Confident Rules Without Support Requirement. In *10th International Conference on Information and Knowledge Management*. pp. 89–96.

Wang, Z., Sun, X. & Zhang, D., 2007. A PSO-based classification rule mining algorithm. In *3rd International Conference on Intelligent Computing*. pp. 377–384.

Webb, G.I., 2001. Discovering associations with numeric variables. In *7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. pp. 383–388.

Webb, G.I., 2000. Efficient search for association rules. In *6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. pp. 99–107.

Wu, X., Zhang, C. & Zhang, S., 2004. Efficient mining of both positive and negative association rules. *ACM Transactions on Information Systems*, 22(3), pp.381–405.

Wu, X., Zhang, C. & Zhang, S., 2002. Mining both positive and negative association rules. In *19th International Conference on Machine Learning*. Sydney, Australia, pp. 658–665.

Xiong, H. & Tan, P.-N., 2003. Mining strong affinity association patterns in data sets with skewed support distribution. In *3rd IEEE International Conference on Data Mining*. pp. 387–394.

Yan, X., Zhang, C. & Zhang, S., 2005. ARMGA: Identifying interesting association rules with genetic algorithms. *Applied Artificial Intelligence: An International Journal*, 19(7), pp.677–689.

Yan, X., Zhang, C. & Zhang, S., 2009. Genetic algorithm-based strategy for identifying association rules without specifying actual minimum support. *Expert Systems with Applications*, 36(2), pp.3066–3076.

Yang, W., Li, D. & Zhu, L., 2011. An improved genetic algorithm for optimal feature subset selection from multi-character feature set. *Expert Systems with Applications*, 38(3), pp.2733–2740.

Ykhlef, M. & ElGibreen, H., 2009. Mining sequential patterns using hybrid evolutionary algorithm. *World Academy of Science, Engineering & Technology*, 60(149), pp.863–870.

Zaki, M.J., 2000. Scalable algorithms for association mining. *IEEE Transactions on Knowledge and Data Engineering*, 12(3), pp.372–390.

Zaki, M.J., 2001. SPADE: An efficient algorithm for mining frequent sequences. *Machine Learning*, 42(1-2), pp.31–60.

Zaki, M.J. & Ogihara, M., 1998. Theoretical foundations of association rules. In *3rd ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*. pp. 1–8.

Zhou, L. & Yau, S., 2007. Efficient association rule mining among both frequent and infrequent items. *Computers & Mathematics with Applications*, 54(6), pp.737–749.